المصادر المفتوحة

خيارات بلا حدود

المهندس

محمد أنس طويلة

المصادر المفتوحة

خيارات بلا حدود

محمد أنس طويلة



مقدمة

خلال السنوات القليلة القادمة، ومع التسارع غير المسبوق في وتيرة التطور التقني، وتحول العالم إلى قرية صغيرة بكل معنى الكلمة، وتبلور مفهوم ما أصبح يدعى بمجتمع الإنترنت، ستجد الحكومات والشركات والأفراد نفسها مضطرة إلى اتخاذ قرارات مصيرية تتعلق باعتمادها على تقنية المعلومات، ومن الضرورة بمكان أن يشترك الجميع في الحوار واتخاذ القرار حول كيفية اعتماد هذه التقنية لما لها من أثر حاسم في تطوير وبناء المجتمع.

نشأت فكرة هذا الكتاب نتيجة عدة حوارات دارت بيني وبين أحد أصدقائي (وأود انتهاز هذه الفرصة في التعبير له عن جزيل شكري وامتناني على تشجيعه المستمر وسعة صبره عندما كان يدفعني على الدوام إلى إتمام هذا العمل رفاعة عكرمة) عندما كنا نتباحث في أهمية المصادر المفتوحة وتأثيرها على مجتمعاتنا العربية، حيث كان يلح علي دوماً لتوثيق ما كنا نتحدث بشأنه، وأنا إذ أقدم هذا العمل المتواضع إلى مكتبتنا العربية أسأل الله أن أكون قد وفقت في تأدية جزء من الدور المناط بي في هذا الحوار، راجياً أن تكون الأفكار الواردة في هذا الكتاب حافزاً يدعو كلاً منا إلى المبادرة والعمل على تفعيل هذا الحوار لتحتل أمتنا العربية الدور الذي يليق بها بين أمم المعمورة.

لقد وضعت نصب عيني منذ البداية أن يكون هذا الكتاب متاحاً لأكبر قدر ممكن من القراء، لأننا جميعاً معنيون ببناء مستقبل أفضل لأوطاننا وأبنائنا، ولذلك فقد حاولت قدر المستطاع أن أعرض المعلومات الفنية بأسلوب سهل مبسط يجعلها قريبة المنال من غير المتخصصين في تقنية المعلومات، وحاولت تجنب التفصيلات التقنية المعقدة والتي قد تحول دون إيصال رسالة هذا الكتاب إلى أوسع شريحة ممكنة من القراء، والتي أرجو أن تكون حافزاً يدفعنا إلى تفهم المصادر المفتوحة ويطلق العنان لإبداعات عربية تمكننا من استثمار كل ما ستحمله لنا هذه الفلسفة الجديدة في صناعة وتطوير البرمجيات من فرص وميزات.

هذا الكتاب، دعوةٌ إلى الحوار والعمل، لنكون منتجين لا مستهلكين في مجتمع المعلومات المقبل، شركاء لا أجراء في بناء الحضارة الإنسانية، أسياداً لا عبيد في غابة العولمة.

محمد أنس طويلة
anas@tawileh.net



قرأت مجدك في قلبي وفي الكتب

شآم... أنت المجد لم يغب

لكل مدينة تاريخ..

إلا دمشق، فمنها يبدأ التاريخ.

إلى مدينتي الرائعة، أهدي هذا العمل المتواضع، رمزاً للعرفان بالجميل.

دمشق في 2004/3/10

الفصل الأول
فلسفة البرمجيّات مفتوحة المصدر

مقدّمة

بشكل عام، تعتبر البرمجيّات مفتوحة المصدر عندما تترافق مع ترخيص يتيح للمستخدم حرية استخدام هذه البرمجيّات لأية غاية كانت، تعديل هذه البرمجيّات أو إعادة توزيع النسخ الأصلية أو المعدّلة منها دون أي مقابل مادي.
تتعارض فلسفة البرمجيّات مفتوحة المصدر حسب التعريف السابق بشكل جذري مع البرمجيّات المغلقة والتي لا يتاح لمستخدمها الإطلاع على شيفرتها المصدريّة أو تعديلها أو إعادة توزيعها، وعلينا هنا ملاحظة الفارق بين البرمجيّات مفتوحة المصدر و البرمجيّات المجانيّة أو المشتركة (والتي لا تترافق عادة مع شيفرتها المصدريّة).

من أشهر أمثلة البرمجيّات مفتوحة المصدر: نظام التشغيل لينكسLinux ، مخدّم الوب أباتشي Apache أما البرمجيّات المغلقة فمن أمثلتها نظام التشغيل ويندوز Windows و مخدّم الوب IIS.

ما هو البرنامج مفتوح المصدر؟

إن مجرد توفر الشيفرة المصدريّة مع أي برنامج لا يكفي لإعتبار هذا البرنامج مفتوح المصدر، و لكي يندرج أي برنامج ضمن إطار المصادر المفتوحة ينبغي أن تتحقق في

إتفاقية الترخيص المرافقة مجموعة من المعايير (لمزيد من المعلومات عن هذه المعايير يمكن مراجعة موقع المصادر المفتوحة www.opensource.org)

تعريف المصادر المفتوحة

النص	غايته
1. حرية إعادة التوزيع يجب ألا تعيق إتفاقية الترخيص أي طرف من بيع أو توزيع البرنامج بالمجان كجزء من برنامج آخر يحتوي وحدات برمجية من عدة مصادر. كما يجب ألا تفرض إتفاقية الترخيص أية رسوم استخدام أو أية رسوم أخرى لقاء هذا التوزيع.	بإلزام إتفاقية الترخيص لإتاحة حرية إعادة التوزيع ستنخفض الحوافز وراء التضحية بالمكاسب بعيدة المدى لقاء عوائد مبيعات مرحلية ضئيلة.
2. الشيفرة المصدريّة يجب أن يحتوي البرنامج على شيفرته المصدريّة كاملة، كما يجب أن تتيح إتفاقية الترخيص توزيع الشيفرة المصدريّة جنباً إلى جنب مع النسخة التنفيذية. وفي حال كان أحد المنتجات يوزع دون شيفرته المصدريّة يجب أن تكون هذه الشيفرة المصدريّة متاحة لمن يود الحصول عليها بسهولة و يسر و بكلفة لا تتجاوز كلفة النسخ أو التوضيب – ويفضل إتاحة هذه الشيفرة المصدريّة عبر	يعتبر الوصول إلى الشيفرة المصدريّة أساسياً كون تطور البرمجيّات يعتمد بالأساس على التعديل المستمر. بما أن الهدف هو جعل التحول سهلاً لا بد من ضمان سهولة التعديل.

الإنترنت بالمجان. وتعتبر الشيفرة المصدريّة الوسيلة الأمثل لتعديل البرنامج من قبل المطورين. ويمنع منعا باتاً تعقيد الشيفرة المصدريّة للبرنامج عمداً، كما يمنع استخدام أية صيغ مرحلية ضمن البرنامج.	
3. الأعمال المشتقّة يجب أن تضمن إتفاقية الترخيص إمكانية إجراء التعديلات و بناء برمجيات جديدة مشتقّة من البرمجيّات الأصلية، كما يجب أن تتيح توزيع هذه البرمجيّات ضمن نفس شروط ترخيص البرنامج الأساسي.	لا يكفي مجرد الإطلاع على الشيفرة المصدريّة لضمان المراجعة المستقلة للبرنامج من قبل أطراف أخرى (وبالتالي ضمان عملية التطوير السريعة والفعّالة). لتحقيق عملية التطوير هذه يجب أن نمكّن المطورين من الوصول بحريّة إلى الشيفرة المصدريّة و تعديلها و إعادة توزيع النسخ المعدّلة.
4. تكامل الشيفرة المصدريّة للمطور الأساسي يمكن أن تمنع إتفاقية الترخيص إعادة توزيع الشيفرة المصدريّة في صيغتها المعدّلة فقط في حال إتاحة توزيع "ملفات التعديل" مع الشيفرة المصدريّة والتي تقوم بتعديل البرنامج أثناء بنائه. كما يجب أن تتيح إتفاقية الترخيص بوضوح توزيع البرامج المبنية	يعتبر تشجيع التعديلات من قبل أطراف متعددة أمراً إيجابياً، إلا أن المستخدم يملك الحق في معرفة المسؤول عن البرمجيّات التي يستخدمها، كما يملك مطورو البرامج الأساسيين الحق في معرفة البرمجيّات التي يطلب إليهم دعمها لحماية سمعتهم. ولذلك فإن ترخيص المصادر المفتوحة يجب أن يضمن إتاحة الشيفرة المصدريّة لأي

على الشيفرة المصدريّة المعدّلة. ويمكن أن تتطلب إتفاقية الترخيص منح البرمجيّات المعدّلة أسماء أو أرقام إصدار تختلف عن تلك المعطاة للبرمجيات الأصلية.	برنامج ينطوي تحت لوائه، ولكن هذا الترخيص قد يفرض ضرورة توزيع الشيفرة المصدريّة الأساسية كقاعدة رئيسية إضافة إلى جميع التعديلات التي تمت إضافتها عبر "برامج التعديل". وهكذا يمكن إجراء أية تعديلات تعتبر "غير رسمية" وجعلها مميزة عن الشيفرة المصدريّة الأساسيّة.
5. لا ينبغي التمييز بحق الأشخاص و المجموعات يجب أن لا تميز إتفاقية الترخيص بحق أي شخص أو مجموعة.	للحصول على الإفادة القصوى من فلسفة المصادر المفتوحة يجب ضمان مساهمة أكبر عدد ممكن من الأشخاص و المجموعات على إختلاف انتماءاتهم ومذاهبهم في تطوير برمجيات المصادر المفتوحة دون أدنى تمييز فيما بينهم. ولذلك يحظر على أية إتفاقية ترخيص للمصادر المفتوحة تحديد أو منع مساهمة أي كان في عملية التطوير و المشاركة. بعض الدول –كالولايات المتحدة الأمريكية– تفرض قيوداً على تصدير مجموعات معينة من البرمجيّات. قد يتم التنويه إلى هذه القيود ضمن إتفاقية الترخيص وإنذار المستخدم أنه قد ينتهك قوانين الدول التي تفرض هذه القيود، إلا أن إتفاقية الترخيص بحد ذاتها لا

	يجوز أن تفرض هذا النوع من القيود على مستخدميها.
6. لا ينبغي أن تميز إتفاقية الترخيص بين أية مجالات لاستخدام البرمجيّات ينبغي ألا تحول إتفاقية الترخيص دون استخدام أي شخص للبرامج مفتوحة المصدر في أي مجال كان. على سبيل المثال يجب ألا تمنع إتفاقية الترخيص استخدام البرنامج في الشركات أو مراكز البحوث ...	الهدف الرئيسية لهذا النص يتجلى في منع أي تلاعب في إتفاقية الترخيص قد يمنع استخدام البرمجيّات مفتوحة المصدر تجارياً. ينبغي دعم وتشجيع المستخدمين للبرمجيات في عالم المال والأعمال على الإنضمام إلى مجتمع المصادر المفتوحة وعدم المساس في حقهم هذا باستثنائهم من هذا المجتمع.
7. توزيع إتفاقية الترخيص يجب أن تنطبق الحقوق و الإمتيازات الملحقة بالبرنامج عبر إتفاقية الترخيص على جميع الأطراف التي يتم توزيع البرنامج إليها دون الحاجة إلى تطبيق أية تراخيص جديدة من قبل هذه الأطراف.	تهدف هذه العبارة إلى منع إغلاق البرمجيّات بأساليب غير مباشرة كإضافة إتفاقية لضمان سرية المعلومات مثلاً.
8. يجب ألا تقتصر إتفاقية الترخيص على منتج محدد يجب ألا تعتمد الحقوق و الإمتيازات الملحقة بالبرنامج بكونه جزءاً من برنامج آخر. إذا كان البرنامج مشتقاً من برنامج أساسي وتمت إعادة توزيعه ضمن نفس الترخيص للبرنامج الأساسي فإن الحقوق والإمتيازات	تمنع هذه العبارة إتفاقيات الترخيص (المضلّلة)

الممنوحة لمستخدمي هذا البرنامج يجب أن تتطابق مع الحقوق الممنوحة لمستخدمي البرنامج الأساسي.	
9. يجب ألا تفرض إتفاقية الترخيص أية قيود على البرمجيّات الأخرى يجب ألا تضع إتفاقية الترخيص أية قيود على البرمجيّات الأخرى المرافقة للبرنامج المرخّص. على سبيل المثال لا ينبغي أن تتضمن إتفاقية الترخيص إشارات لحصر البرمجيّات المرافقة للبرنامج المرخص بالبرمجيّات مفتوحة المصدر دون غيرها.	يملك موزعي البرمجيّات مفتوحة المصدر مطلق الحرية في تقرير التركيبة الملائمة لحالة كل منهم على حدة. يتطلب ترخيص GPL من البرمجيّات المرتبطة مع وحدات برمجية مشمولة بترخيص GPL أن تنطوي ضمن نفس الترخيص فقط في حال كانت هذه الوحدات البرمجية و البرمجيّات تشكل عملاً متكاملاً و ليس في حال مجرد وجود هذه الوحدات البرمجية إلى جانب البرمجيّات الأخرى دون أدنى ترابط فيما بينها.

يحاول تعريف المصادر المفتوحة التعبير عن المعنى الحقيقي لتعبير "المصادر المفتوحة" كما تريد الغالبية العظمى من مطوري البرمجيّات، إلا أن هذا التعبير ومع الإنتشار الواسع لإستخدامه قد فقد جزءا من الدقة في تعريفه.

يعتقد الكثيرون أن فلسفة المصادر المفتوحة تنطوي على فوائد عدة من أهمها تسريع تطور البرمجيّات و إنتاج برمجيات ذات جودة أعلى من تلك التي تطور ضمن أساليب المصادر المغلقة.

مبادرة المصادر المفتوحة Open Source Initiative (OSI)

مبادرة المصادر المفتوحة هي مؤسسة لا تهدف إلى الربح (www.opensource.org) أنشئت بغاية إدارة ونشر تعريف المصادر المفتوحة بما يعود على الفائدة على المجتمع بأسره. وتقوم مبادرة المصادر المفتوحة بمهامها بشكل رئيس عبر برنامج وعلامة الترخيص التي قامت بتطويرها. سنقوم فيما يلي بشرح تفاصيل برنامج علامة الترخيص لمبادرة المصادر المفتوحة.

برنامج علامة ترخيص المصادر المفتوحة

يحدد تعريف المصادر المفتوحة الخصائص الأساسية للبرمجيات ذات المصادر المفتوحة, ولكن تعبير "المصادر المفتوحة" بحد ذاته قد يكون عرضة لإساءة الإستخدام، ولأن هذا التعبير وصفي بالمعنى العام "أي أنه يصف شيئاً معيناً" لذلك لا يمكن حمايته كعلامة تجارية.

ولأن المستخدمين بحاجة إلى طريقة موثوقة لمعرفة ما إذا كان برنامج ما ينطوي تحت إطار المصادر المفتوحة بالفعل فقد طوّرت مبادرة المصادر المفتوحة "علامة ترخيص

المصادر المفتوحة". تنطبق علامة ترخيص المصادر المفتوحة على البرمجيّات وليس على إتفاقية الترخيص. ويستطيع مطورو البرمجيات إستخدام علامة ترخيص البرمجيّات مفتوحة المصدر عند الحاجة دون الرجوع إلى مبادرة المصادر المفتوحة. وتتطلب عملية إستخدام الترخيص الخطوات التالية:

- تقوم مبادرة المصادر المفتوحة بإعداد لائحة تضم جميع إتفاقيات الترخيص التي تتوافق مع تعريف البرمجيّات مفتوحة المصدر. تقوم المبادرة بإستقبال جميع إتفاقيات الترخيص والتي يستطيع أي شخص التقدم بها بغاية الحصول على موافقة المبادرة عليها وتنتخب منها تلك الإتفاقيات التي تتوافق مع تعريف المصادر المفتوحة و تضيفها إلى لائحتها. وإذا كنت ترغب بإضافة إتفاقية ترخيص جديدة بإمكانك الإتصال بالمبادرة على عنوان البريد الإلكتروني التالي: license-approval@opensource.org
- إذا كنت ترغب بإستخدام علامة ترخيص المصادر المفتوحة مع برمجياتك فبإمكانك إختيار إتفاقية ترخيص حائزة على موافقة مبادرة المصادر المفتوحة وتوزيعها مع برمجياتك.

يعتبر الحصول على موافقة لإتفاقية ترخيص جديدة أمراً سهلاً للغاية، وتقدّم مبادرة المصادر المفتوحة مجموعةً من الإرشادات والتعليمات الخاصة بذلك ضمن موقعهم على الإنترنت: www.opensource.org/docs/certification_mark.html

إستخدام علامة الترخيص

يستطيع المطورون إستخدام علامة ترخيص المصادر المفتوحة على أي برنامج يوزع ضمن إتفاقية ترخيص حائزة على موافقة مبادرة المصادر المفتوحة. ولتمييز البرمجيّات التي تحمل علامة ترخيص المصادر المفتوحة يمكن أن يضيف المطورون إلى هذه البرمجيّات إحدى الجملتين الآتيتين شريطة عدم تعديلهما:

هذا البرنامج برنامج مفتوح المصدر مرخص من مبادرة المصادر المفتوحة
This software is OSI Certified Open Source Software

علامة ترخيص المصادر المفتوحة هي علامة ترخيص من مبادرة المصادر المفتوحة
OSI Certified is a certification mark of the Open Source Initiative.

والجملة الأخرى:

برنامج مفتوح المصدر مرخص من مبادرة المصادر المفتوحة
OSI Certified Open Source Software.

تملك كل توزيعة من برمجياتك متطلباتها الخاصة:

- إذا كنت ترغب بتوزيع البرنامج بصيغة إلكترونية بإمكانك إضافة التنويه إلى ملف README أو أي ملف آخر تتوقع أن يكون أول ما سيقوم أي مستخدم بقراءته.
- إذا كنت ترغب بتوزيع برنامجك بصيغة فيزيائية يتوجب عليك القيام بما يلي:
  - إذا كان البرنامج مرفقاً مع أية مواد مطبوعة يجب طباعة التنويه بشكل واضح على هذه المواد.
  - إذا كان البرنامج موزعاً ضمن أقراص ضوئية أو مغناطيسية أو شرائط مغناطيسية حيث يمكن إضافة إحدى العبارتين على الأقل بطريقة لا تؤذي أداء الأقراص أو الشرائط بحيث تتمكن العين البشرية من قراءتها

بسهولة، عندئذ يتوجب إضافة إحدى العبارتين على هذه الأقراص أو الشرائط.

- إذا كان المواد التي تحتوي على البرنامج موزعة ضمن علبة تمنع قراءة التنويه المطبوع على المواد نفسها يتوجب عندئذ إضافة التنويه بالكامل على السطح الخارجي للعلبة.

إذا لم تنطبق أي من التعليمات الآنفة الذكر على حالتك عليك الإتصال بمبادرة المصادر المفتوحة ليقوموا بدورهم بإضافتها إلى هذه القائمة.

البرمجيّات مفتوحة المصدر في مواجهة البرمجيّات المغلقة

1. إعتبارات عامة

هناك العديد من الإعتبارات غير الفنية التي يجب أخذها بعين الإعتبار عند إختيار الحل البرمجي الأمثل للإستخدام في ظروف معينة. تتضمن هذه الإعتبارات عادة ما يلي:

- إمكانية التطور

وهو يعني إمكانية تشغيل نفس النظام البرمجي لكل من المشاريع الصغيرة والكبيرة، وتمتاز البرمجيّات مفتوحة المصدر في هذا السياق كون ترافقها مع الشيفرة المصدريّة يتيح تعديلها بما يتلاءم مع غالبية البيئات والمشاريع.

- كلفة التطوير الكليّة

وهي تعتبر معياراً فائق الأهمية، فقد يبدأ منتج معين بكلفة بدائية منخفضة ولكنّه على المدى البعيد سيكلّف مبالغ أكبر بكثير من الإستثمارات الأولية.

هذه الإعتبارات فائقة الحساسية لتغيرات مجموعة من الإفتراضات التي توضع وفقاً لكل حالة على حدة، لذلك ينبغي تحديد جميع المعايير المؤثرة في التكاليف الكليّة وتقدير النفقات المرتبطة بكل منها بدرجة مقبولة من الدقة.

هذه الإعتبارات قد تتضمن:

- تكاليف الإدارة والمتابعة.
- تكاليف التطوير.
- تكاليف الدعم الفني.
- خيارات التجهيزات المتاحة.

إعتبارات أخرى

أهم الأمور التي تواجه غالبية المستخدمين هي:

- الحيطة و الحذر من الأخطار والمحدودية المترافقة مع الحلول ذات المصدر الوحيد.
- الحيطة والحذر من مشاكل إدارة التراخيص والأمور القانونية. (حيث يجني مصنّعوا البرمجيّات المغلقة أرباحهم عبر بيع تراخيص الإستخدام).
- المرونة: بإمكان مستخدمي البرمجيّات مفتوحة المصدر تعديل البرمجيّات وفق الحاجة لتلبّي متطلباتهم بطرق يستحيل تحقيقها دون توفر الشيفرة المصدريّة.

مخاوف غير مبررة

يتجنب البعض إستخدام البرمجيّات المفتوحة المصدر بسبب مخاوف غير مبررة منها:

تفنيده	التخوف
يوجد فعلياً نوعان من الدعم الفني للبرمجيات المفتوحة المصدر: الدعم الفني التقليدي المدفوع القيمة و الدعم غير الرسمي المتوفر عبر مجموعات المستخدمين لهذه البرمجيّات، وهناك أشكال عدة لكل من هذين النوعين.	عدم توفر الدعم الفني اللازم
تتمحور بعض الإنتقادات للبرمجيات مفتوحة المصدر حول عدم توفر ضمانات قانونية كافية في حال الضرورة للجوء للتحكيم أو الحكم القانوني. ولكن في المقابل تمنع جميع إتفاقيات ترخيص البرمجيّات المغلقة المستخدم من محاكمة الشركة المنتجة، ومن كان يعتقد أن بإمكانه محاكمة شركة عملاقة كـــ (مايكروسوفت) في حال وقوع أي إشكال فهو يسرح في شطحات من ابتكار خياله.	عدم توفر ضمانات للحقوق القانونية
إذا توقف المطور الأساسي لسبب ما أو لآخر عن توفير الدعم لبرنامجه يمكن عندئذ لأي شخص أو مجموعة المبادرة لتقديم الدعم عوضاً عنه، وقد أثبتت البرمجيّات مفتوحة المصدر تفوقاً واضحاً في هذا السياق.	المخاطرة بإحتمال توقف تطوير البرنامج

يبدو أن البرمجيّات مفتوحة المصدر غير مجدية إقتصادياً وقد تختفي بين ليلة وضحاها	هناك الكثير من الشركات التي تجني أرباحها من البرمجيّات مفتوحة المصدر مثل ريدهات RedHat. تعتمد هذه الشركات على توفير الدعم الفني والخدمات الإستشارية وأدوات التطوير لجني الأرباح.
فلسفة برمجيات المصادر المفتوحة تتنافى مع جميع حقوق الملكية الفكرية	يمكن إستخدام البرمجيّات ذات المصادر المفتوحة (كبرامج تحرير النصوص) لتطوير أية معلومات خاصة أو محمية، ويمكن الإبقاء على هذه المعلومات الخاصة محمية دون أدنى حاجة لإعادة توزيعها تحت إتفاقيات ترخيص المصادر المفتوحة.

إتفاقية ترخيص GPL

تعتبر إتفاقية ترخيص GNU Public License (GPL) حجر الأساس في عالم المصادر المفتوحة، وقد ظهرت للمرة الأولى عبر مؤسسة تدعى GNU is Not Unix (GNU) – ولا يعلم أحد المعنى الأساسي المقصود من هذا الإسم، إلا أنه ممتع للغاية، وقد سلكت العديد من البرمجيّات مفتوحة المصدر نفس الطريقة الطريفة في تسميتها– والتي تقوم بتطوير العديد من حزم البرمجيّات. تأسست GNU من قبل ريتشارد ستولمان Richard Stallman (الملقب بأبو البرمجيّات مفتوحة المصدر) في عام 1984 وعملت على تطوير إتفاقية ترخيص مخصصة لضمان إستمرارية نجاح حركة البرمجيّات مفتوحة المصدر. (للمزيد من المعلومات يمكن مراجعة موقع GNU على الإنترنت www.gnu.org)

أكثر الخصائص أهمية في إتفاقية ترخيص GNU هو أن هذه الإتفاقية لا تحمي شخصاً أو مؤسسةً بعينها، بل تقوم بحماية الشيفرة المصدريّة للبرنامج. لقد مكّنت قوانين حفظ حقوق الملكية الفكرية في الماضي أي شخص إدعاء ملكية برنامج ما ومن ثم بيعه مقابل ربح مادّي. كما منحت قوانين حماية الملكية الفكرية هذا الشخص الحق بالإدعاء على أي شخص يستخدم نفس البرنامج لبناء تطبيقات مماثلة.
قام ريتشارد ستولمان Richard Stallman و إيريك ريموند Eric Raymond وغيرهما بإنشاء ونشر ثقافة جديدة في عالم البرمجيّات تجلّت في إتفاقية ترخيص البرمجيّات مفتوحة المصدر GPL.

تتيح إتفاقية ترخيص GPL تعديل، نسخ وتوزيع البرمجيّات التي تستخدمها شريطة إستخدام البرمجيّات المعدّلة لترخيص GPL وتوفر الشيفرة المصدريّة معها. أي أن هذه الإتفاقية ستضمن للمطور الأساسي لأي برنامج أن برنامجه سيبقى مفتوح المصدر مهما تم تعديله وتطويره. يمكن إستخدام الشيفرة المصدريّة الموزّعة ضمن إتفاقية ترخيص GPL لبناء برمجيات جديدة أو معدّلة، ولكن لا يمكن تقييد حقوق الآخرين من الحصول على نفس الحقوق المتوفرة مع البرنامج الأساسي.

شأنها شأن جميع الوثائق القانونية تصطدم إتفاقية الترخيص GPL ببعض العوائق فيما يتعلق بالتعريفات للمصطلحات المذكورة ضمنها، فما الذي يحدد الفرق بين الشيفرة المصدريّة المعدّلة أو الشيفرة المصدريّة الجديدة (الأساسية)؟

لنلاحظ ما يلي:

- يتم تعديل الشيفرة المصدريّة للبرمجيات المفتوحة المصدر لإضافة ميزات جديدة وتصبح عندها تلك البرمجيّات (معدّلة)
- عندما يتم تطوير البرنامج من قبل مجموعة من المطورين تزداد الأمور تعقيداً، بشكل عام يعتبر البرنامج (معدّلاً) عندما يكون مشتقاً بشكل مباشر من برنامج آخر أو مرتبطاً مع برنامج آخر. (ولا يهم ما إذا كان هذا الإرتباط ساكناً أم ديناميكياً، بمعنى أن إستخدام أية وحدة برمجية يجعل البرنامج بأكمله برنامجاً معدّلاً من هذه الوحدة).

على العكس مما قد يتبادر إلى ذهنك للوهلة الأولى فإن البرمجيّات مفتوحة المصدر المحمية بإتفاقية ترخيص GPL ليست بالضرورة مجانيّة، بمقدور أي شخص أو شركة الحصول على برنامج مفتوح المصدر وتعديله ومن ثم بيعه، ولكن يتوجب على هذا الشخص أو الشركة توفير الشيفرة المصدريّة مع هذا البرنامج لإتاحة الإطّلاع عليها أو تعديلها لمن يشاء.

إتفاقية ترخيص LGPL

وهي إتفاقية لترخيص البرمجيّات مفتوحة المصدر المستخدمة عادة لترخيص إستخدام المكتبات البرمجية لإتاحة ربط هذه المكتبات بالبرمجيّات غير مفتوحة المصدر، من أهم المكتبات المرخصة بإتفاقية ترخيص LGPL مكتبة لغة البرمجة C (Glibc) مما يبرز أهمية هذه الإتفاقية بالنسبة لنظام التشغيل لينكس.
تتوفر مكتبة Glibc كمكتبة مشتركة ضمن معظم توزيعات لينكس، وهي تقدم قناة للتواصل بين غالبية تطبيقات لينكس ونواة نظام التشغيل. ونعني بكلمة مشتركة هنا أن البرمجيّات التي ستستخدم هذه المكتبة سترتبط بها أثناء تشغيلها.

دراسة حالة

مخدّم الوب أباتشي Apache Web Server

نشأة أباتشي

في عام 1995 كان أكثر مخدّمات الوب شعبية واستخداماً على الإنترنت هو مخدّم الوب المجاني HTTP Daemon والذي قام بتطويره روب ماك كول Rob MacCool من المركز الوطني للتطبيقات الحاسوبية الفائقة National Centre for Supercomputing Applications في جامعة إيللنويز.

لكن تطوير المخدّم httpd توقف بعد ترك روب للمركز في منتصف عام 1994 وبعدها قام العديد من المطورين بالكثير من الإضافات والتعديلات على البرنامج الأساسي ولكنها بقيت مشتتة دون دمجها في نسخة موحدة. وبغية تنسيق هذا الكم من الإضافات والتعديلات شكل هؤلاء المطورون مجموعة صغيرة تألفت من ثمانية أعضاء أساسيين أسميت مجموعة أباتشي Apache Group.

أسس أعضاء مجموعة أباتشي في عام 1999 مؤسسة أباتشي Apache Software Foundation لتقدم الدعم التنظيمي، القانوني و المادي لمخدّم الوب أباتشي.

فلسفة أباتشي

يعتمد أباتشي على فلسفة المصادر المفتوحة، ويعتبر من أكثر البرمجيّات المفتوحة المصدر إنتشاراً في العالم.

تتميز مشاريع أباتشي بمنهجية تطوير جماعية وإتفاقية ترخيص مفتوحة وإرادة لبناء برامج ذات نوعية عالية تحتل دوماً موقع الريادة. ويرجع الفضل إلى هذه الفلسفة في جعل أباتشي أكثر مخدّمات الوب شعبية على الإنترنت منذ شهر نيسان عام 1996 متمتعاً بنسبة إستخدام تفوق جميع مخدّمات الوب الأخرى مجتمعة.

منهجية أباتشي

إتفاقية الترخيص الخاصة بمخدّم الوب أباتشي مختصرة ومباشرة، وتحتوي على خمسة نقاط أساسية:

- بإمكان المستخدم إعادة توزيع البرنامج سواء بصيغته الأساسية أو مع أية تعديلات شريطة إرفاقه مع تنويه حفظ الملكية وشروط الترخيص الملائمة.
- يجب إرفاق التحذير مع أية نسخة.
- يجب ذكر كلمة أباتشي Apache تحديداً.
- يحظر إستخدام كلمة أباتشي Apache لتسويق أية منتجات مشتقة دون إذن مسبق.
- يحظر إستخدام كلمة Apache ضمن أية منتجات مشتقة.

وكما في كل البرمجيّات ذات المصادر المفتوحة فإن الغاية من هذا الترخيص تكمن في إتاحة إستخدام وتعديل ونسخ أباتشي من قبل أي شخص شريطة تمرير نفس هذه الحقوق إلى الأطراف الأخرى.

أسئلة عليك إجابتها

إذا كنت مديراً أو مسؤولاً وتفكر في الإنتقال إلى عالم البرمجيّات مفتوحة المصدر تأكد من استيعابك خصائص البرمجيّات مفتوحة المصدر، فقد لا تكون هذه البرمجيّات هي الحل الملائم لمتطلباتك. عليك أن تسأل نفسك الأسئلة التالية:

- هل أملك الوقت والموارد الكافية لتدريب المستخدمين على هذه البرمجيّات؟
- هل هذه البرمجيّات موثوقة إلى الحد الذي يسمح لي باستخدامها؟
- كيف سأقوم بمراجعة الشيفرة المصدريّة للتحقق من ملاءمتها أمنياً؟
- كيف سأبرر قرار التحول إلى المصادر المفتوحة لمدرائي؟
- كيف سأشرح قرار إستخدام المصادر المفتوحة لزبائني وشركائي؟

الفصول اللاحقة من هذا الكتاب ستحاول الإجابة على هذه الأسئلة لتساعدك على إتخاذ القرار الصحيح حول التحول إلى برمجيات المصادر المفتوحة.

الفصل الثاني

تعريف بنظام التشغيل لينكس

لمحة تاريخية

يعتبر يونيكس Unix من أكثر أنظمة التشغيل إنتشاراً على مستوى العالم، وقد تم تطويره أساساً كنظام تشغيل متعدد المهام للمخدّمات والحواسيب الضخمة في منتصف السبعينيات لينتشر إستخدامه بعدها ويصبح من أشهر أنظمة التشغيل.

لقد منح يونيكس المستخدمين طريقة معيارية للتخاطب و التحكم بالحواسيب. لكن هذه المعايير لم تكن (معيارية) كفايةً بالمعنى الدقيق للكلمة، فقد قام العديد من مصنعي التجهيزات بإنتاج إصدارات خاصة من يونيكس بغاية حصر المستخدمين بمنتجاتهم و كان المعيار الوحيد الذي يجمع كل هذه الإصدارات هو السعر الفلكي لأي منها. و مع تطور صناعة تكنولوجيا المعلومات و انخفاض أسعار التجهيزات بشدة، لم تتأثر أسعار أنظمة التشغيل يونيكس بهذا الإنخفاض مما أدى إلى احتكار شركة وحيدة كبرى لأسواق أنظمة التشغيل المكتبية (وسأترك لك تقدير اسم هذه الشركة الكبرى).

في عام 1991 بدأ طالب شاب من جامعة هلسنكي بفنلندا (لينوس تورفالدز Linus Torvalds) ببناء نظام تشغيل جديد متوافق مع يونيكس ليكون هذا النظام الجديد متاحاً عبر الإنترنت مجاناً كبرنامج مفتوح المصدر (البرمجيات المفتوحة المصدر هي البرمجيات التي توزع معها شيفرتها المصدرية Source Code).

بسرعة هائلة، توافدت أعداد كبيرة من المبرمجين و المطورين من جميع أصقاع العالم و شكّلوا مجتمعاً افتراضياً تبنّى تطوير النظام الجديد (والذي دعي لينكس نسبةً إلى واضع لبنته الأساسية لينوس تورفالدز) وأخذ هذا النظام بالتطور وازداد تقدماً وتعقيداً. و على الرغم من كون الإصدارات الأولى من لينكس موجهةً بشكل رئيس إلى المبرمجين والمطورين فقد قام هؤلاء ببناء برمجيات تساعد المستخدمين العاديين وغير المتخصصين في دخول عالم لينكس. وتقدر نسبة مستخدمي لينكس حالياً بتسعة في المائة من جميع مستخدمي الحواسيب في العالم، و هذه النسبة تتزايد بمعدلات تفوق أي نظام تشغيل آخر.

ترصد التوقعات أعداد مستخدمي لينكس في العالم بما يقارب 20 مليون مستخدم (وأعتقد أن هذا الرقم سيكون صغيراً للغاية عند قراءتك هذه السطور).

لم يكن للينكس أن يرى النور دون استخدام أدوات التطوير التي وفرتها مؤسسة البرمجيات المجّانية(Free Software Foundation) والتي تعرف بإسم GNU. لقد تداخلت أدوات GNU في جميع مراحل تطوير لينكس منذ بداياتها الأولى. ونظراً للمساهمة الأساسية لهذه الأدوات في بناء لينكس فقد طلبت مؤسسة البرمجيات المجانية إطلاق إسم GNU/Linux على توزيعات لينكس.

يعتبر لينكس اليوم نسخةً كاملة عن يونيكس، وبمقدوره تشغيل أية برمجيات تعمل ضمن يونيكس. كما تم تعديل معظم البرمجيات الحرّة لتعمل ضمن لينكس، وتقوم الشركات التجارية أيضاً بإنتاج إصدارات من برمجياتها لتعمل ضمن لينكس. وتوفّر الإصدارات الجديدة من لينكس دعماً لأطياف أوسع من التجهيزات. وقد أجريت العديد من إختبارات الأداء على لينكس والأنظمة المصادر المفتوحة وأظهرت النتائج تفوّق لينكس الواضح عل

غالبية أنظمة التشغيل الأخرى. من كان بمقدوره أن يتصور أنّ هذه النسخة (المتواضعة) ليونيكس ستغيّر معالم تقنية المعلومات في العالم بأسره وتقلب مفاهيم صناعة تقنية المعلومات رأساً على عقب؟

لينكس مجاني –بالكامل!

إن الأمر الثوري في لينكس – عدا عن حقيقة أنه لن يتوقف عن العمل كل خمس دقائق– و الذي يستحق الكثير من النقاش و لكننا سنحاول إيجازه قدر الإمكان هو طريقة توزيعه. ففي حين تتقاضى شركات البرمجيات الكبرى مبالغ طائلة من زبائنها لقاء استخدامهم لبرمجياتها فإن لينكس مجاني بكل ما تحمل هذه الكلمة من معنى! تستطيع مثلا الحصول على لينكس عبر الإنترنت (هذا في حال كنت من القلائل الذين يملكون سعة من الصبر لتحمل الأزمان الفلكية – وجيوبك كبيرة بما يكفي لتحمل فواتير الإنترنت المقابلة– التي سيتطلبها ذلك عبر مزودي خدمة الإنترنت لدينا!)، كما تستطيع نسخه من أحد أصدقائك دون أن تخرق قوانين حفظ الملكية الفكرية أو تعرض نفسك للملاحقة القانونية و التي غالبا ما ترهبك بها شركات البرمجيات العالمية.

على الرغم من مجانية لينكس فإن ذلك لا يمنع أن البعض قد يجد مكانا لتحقيق بعض المكاسب المادية، لنذكر على سبيل المثال بعض الشركات (و التي يتزايد عددها يوما بعض يوم) التي تأسست لتقديم (توزيعات Distributions) مختلفة من لينكس و التي تقدم للمستخدمين بالإضافة إلى نظام التشغيل المميز عدداً كبيراً من الأدوات و البرمجيات و بيئات التطوير الإضافية. كما تقدم هذه الشركات خدمات الدعم الفني و الخدمات الإستشارية.

في عالم توزيعات لينكس تضيء أسماء كبيرة كـــ RedHat, Suse, Debian (وهي مجرد لمحة عن العدد الهائل من الشركات التي يشكل لينكس نواة أعمالها التجارية) و لتلقي بنفسك نظرة على الكم الهائل لهذه الشركات ما عليك سوى استخدام أحد محركات البحث على الإنترنت للبحث عن "توزيعات لينكس" و كن مستعدا لكم النتائج المذهل! تتنوع توزيعات لينكس من حيث الكم والنوع، ولكنها بعامّتها تأتي مع تشكيلة واسعة من البرمجيات المجانية أو مفتوحة المصدر والتي تستطيع تطويعها وتعديلها لتلائم متطلباتك الخاصة. كما تحتوي على مجموعة من الأدوات التي ستساعدك على تثبيت وإعداد لينكس عبر واجهة رسومية سهلة وبسيطة.

إتفاقية ترخيص لينكس

يوزّع لينكس تحت إتفاقية الترخيص العمومية GNU Gneral Public License أو GPL والتي طوّرت في الأساس من قبل مؤسسة البرمجيات الحرّة لمشروع GNU. وتحدد هذه الإتفاقية شروط توزيع وتعديل البرمجيات الحرّة.

لابد من التنويه هنا إلى الخطأ الشائع بأن البرمجيات المشمولة بترخيص GNU هي برمجيات "مجانية"، وسبب هذا اللغط هو الترجمة الخطأ لكلمة Free من اللغة الإنكليزية إلى العربية، ففي هذا السياق لا يقصد بهذه الكلمة معنى "المجانية" وحسب، بل تعني "الحرّية" (من المصدر الإنكليزي Freedom).
وليت إساءة فهم ترخيص GPL تقف عند هذا الحد، ولذلك سنحاول فيما يلي تبسيط هذا الموضوع لإزالة اللبس في تفسير ترخيص GPL.

في البداية، قام لينوس تورفالدز بتوزيع لينكس ضمن إتفاقية ترخيص أكثر حزماً من إتفاقية GPL والتي تنص على إتاحة توزيع وتعديل البرمجيات بحرّية ولكنها منعت أية تعاملات مادّية في هذا السياق. أمّا إتفاقية GPL فهي تتيح بيع البرمجيات الحرّة وتحقيق الأرباح من وراء ذلك، ولكنها تمنع تحديد حقّ الآخرين في توزيع هذه البرمجيات بأية صيغة.

علينا في البداية أن نوضّح بأن البرمجيات الحرّة المشمولة بإتفاقية الترخيص GPL ليست ملكية عامّة. البرمجيات ذات الملكية العامّة هي برمجيات غير مرخّصة ومتعود ملكيتها لعامّة الناس. أما البرمجيات المشمولة بإتفاقية ترخيص GPL فحقوقها محفوظة لكاتبها، هذا يعني أنّ هذه البرمجيات محمية بقوانين حماية الملكية الفكرية الدولية وبأن كاتب أو صاحب هذه البرمجيات معرّف قانونياً. ومجرد كون البرنامج متاحاً للتوزيع بحرّية لا يعني أنه أصبح ملكاً عاماً.

بإمكان الأشخاص والمؤسسات توزيع البرمجيات المشمولة بإتفاقية الترخيص GPL لقاء مقابل مادّي وبإمكانهم أيضاً تحقيق الأرباح من بيع البرمجيات الحرّة وتوزيعها. ولكن عند بيع البرمجيات المشمولة بإتفاقية الترخيص GPL لا يستطيع الموزّع تجريد المستخدم من الحقوق التي تكفلها له الإتفاقية، على سبيل المثال إذا قمت بالحصول على برنامج مشمول بإتفاقية ترخيص GPL لقاء مقابل مادّي من مصدر ما بإمكانك إعادة توزيع هذا البرنامج مجاناً أو لقاء مقابل مادّي أيضاً.

قد يبدو هذا النقاش متناقضاً للوهلة الأولى، فلماذا تباع البرمجيات لقاء مقابل مادّي لجني الأرباح في حين تتيح إتفاقية الترخيص GPL الحصول على هذه البرمجيات مجاناً؟ عندما تقوم شركة أو مؤسسة ما بتجميع عدد كبير من البرمجيات على قرص مدمج لتوزيعه قد

تحتاج إلى مقابل مادّي لتغطية هذه النفقات، وقد تقرر هذه الشركة أو المؤسسة أنها تريد جني الأرباح لقاء خدماتها، وإتفاقية الترخيص GPL تتيح ذلك.

يتوجّب على المؤسسات التي تقوم ببيع البرمجيات الحرّة الإلتزام بعدّة معايير أساسية محدّدة ضمن إتفاقية ترخيص GPL: أولا، لا يجوز لهذه المؤسسات تجريد المستخدم الذي سيشتري هذه البرمجيات من حقّوقه المنصوص عنها في الإتفاقية. فإذا قمت بشراء قرص مدمج يحتوي برمجيات مشمولة ترخيص GPL فإنك تستطيع نسخ وتوزيع هذا القرص المدمج مجاناً، بل تستطيع أنت أيضاً بيعه. ثانياً، ينبغي أن يوضّح الموزعون لزبائنهم أن هذه البرمجيات مشمولة بإتفاقية ترخيص المصادر المفتوحة. ثالثاً: يجب أن يرفق الموزعون هذه البرمجيات بشيفرتها المصدرية مجاناً ودون أي مقابل مما يتيح لأي شخص سيشتري البرمجيات المشمولة بإتفاقية ترخيص GPL تعديل هذه البرمجيات.

إن السماح للشركات والمؤسسات بيع وتوزيع البرمجيات الحرّة أمر جيد وهام للغاية. فالوصول إلى الإنترنت غير متاح لكل الناس، وحتى عند توفره فقد تكون تكاليف الإتصال خلال المدّة الكافية للحصول على البرنامج المطلوب باهظة. ولذلك تتيح إتفاقية ترخيص GPL للشركات بيع البرمجيات الحرّة للمستخدمين الذين لن يتمكنوا من الحصول عليها عبر الإنترنت (نظراً للتكاليف الباهظة المرافقة). هناك العديد من الشركات التي تقوم ببيع لينكس على أقراص مدمجة أو شرائط ممغنطة وتجني قدراً من الأرباح مقابل خدماتها. هذه الأرباح قد لا تصل إلى مطوري لينكس على الإطلاق، وهو ما قد وافق عليه المطورون عندما أدرجوا برمجياتهم ضمن إتفاقية ترخيص GPL. بمعنى آخر، لقد أدرك لينوس بأن الشركاتا التجارية قد تبادر إلى بيع لينكس بغاية جني الأرباح دون أن يحصل هو شخصياً على أية حصة من هذه الأرباح.

لا تشكل العوامل المادّية أهمية في عالم المصادر المفتوحة مقابل الهدف الأساسي لها وهو تطوير وتوزيع برمجيات متميزة فائقة الجودة وتوفيرها للجميع دون أدنى إستثناء.

ماذا لو قرر لينوس تورفالدز تغيير ترخيص لينكس؟

هذا هو المستحيل بعينه! قانونياً يتطلب تغيير الترخيص الذي يوزع لينكس وفقه من GPL إلى أي ترخيص آخر الحصول على موافقات جميع الأشخاص الذين تبرعوا بتطوير الشيفرة المصدرية للينكس (والذين يتجاوز عددهم عدة مئات). ينتمي هؤلاء الأشخاص إلى دول مختلفة، مما يجعل هذه العملية غير مجدية عملياً. وحتى لو تحقق ذلك فإن مجموعة جديدة من المطورين ستتقدم وتحمل لواء تطوير الشيفرة المصدرة للينكس مستخدمين آخر إصدارة متوفرة قبل تغيير الترخيص وسيتابعون العمل للإبقاء على هذه النسخة (الحرّة) المعيار المعتمد لأنظمة التشغيل مفتوحة المصدر، سواء أحب لينوس ذلك أم لا.

ماذا لو توقف لينوس تورفالدز عن دعم لينكس؟

هناك الكثير من مطوري لينكس ممن يملكون الخبرات الكافية للقيام بمهام لينوس، وعلى الأغلب ستقوم مجموعة من المطورين الماهرين بأداء هذه المهام في حال تنازل لينوس عنها. وقد تنقسم هذه المجموعات إذا ظهرت خلافات فيما يتعلق بالقضايا البرمجية وبالتالي قد يتقسم لينكس نفسه، وقد تعود هذه المجموعات إلى الإتحاد من جديد. ولا داعي للقلق من هذا الإحتمال، فقد مر العديد من مشاريع البرمجيات مفتوحة المصدر بنفس هذه التجربة دون أن تشكل أي تهديد سلبي، وعلى الأخص من وجهة نظر المستخدم، فمجرد كون هذه

البرمجيات مفتوحة المصدر فإن ذلك سيضمن تمتعها بقدر كاف من التطوير والدعم لتلبية متطلبات مستخدميها. وعلى الأغلب فإن المستخدم سيختار توزيعةً شهيرةً للينكس قام بإعدادها أشخاص أمعنوا التفكير في جميع هذه القضايا.

توزيعات لينكس

في عالم المصادر المفتوحة لا حاجة لك كمستخدم للحصول على البرمجيات عشوائياً من عدة مصادر مختلفة كما كنت تفعل مع ويندوز، فعندما ستحصل على لينكس سيأتيك مضمناً في "توزيعة" قياسية، غالباً ما توزّع بصيغة أقراص مدمجة CD. تحتوي هذه التوزيعة بالإضافة إلى نظام التشغيل لينكس مجموعة كبيرة من التطبيقات والأدوات البرمجية التي تم إختيار وتجربة وتفحص كل منها لتتناغم مع البرمجيات الأخرى المضمّنة في التوزيعة، وهذه هي المسؤولية التي تطوع منتجوا توزيعات لينكس لتحمّلها.

بعكس ما يعتقده البعض، لا تسيطر أية شركة على لينكس، وجميع الشركات التي تقوم بتوزيع لينكس تفعل ذلك إنطلاقاً من مبادرات خاصة، وذلك لا يعني إطلاقاً أن شركة دون أخرى تملك الحق في توزيع لينكس بشكل حصري. بإمكان أية شركة ترغب في دخول سوق توزيعات لينكس تجميع مكونات التوزيعة الخاصة بها وإنتاجها وتسويقها شريطة إحترامها لإتفاقية ترخيص المصادر المفتوحة.

لن تتم إضافة أي برنامج إلى توزيعة من توزيعات لينكس ما لم تتم تجربة هذا البرنامج وإستخدامه بشكل كاف لتقييم أدائه وأهميته، وبالتالي فإن توزيعات لينكس ستضمن لك الحصول على البرمجيات التي تحقق الحد الأدني من معايير الجودة بالنسبة للمستخدم.

تتشابه توزيعات لينكس المختلفة فيما بينها بشكل كبير، وهي متوافقة أيضاً من حيث قدرتها على تشغيل التطبيقات البرمجية (بمعنى أن البرنامج المعدّ للعمل ضمن توزيعة ما سيعمل ضمن التوزيعات الأخرى). لنلاحظ هنا أن الإصدارات المختلفة من يونيكس لا تتمتع بهذه الميزة. الحالة الوحيدة التي ستحول دون إمكانية تشغيل التطبيقات البرمجية على توزيعات أخرى هي تغير بنية التجهيزات المستخدمة (كنوع المعالج مثلاً)، مع ذلك فإن توفر الشيفرة المصدرية مع البرمجيات مفتوحة المصدر سيتيح إعادة بناء هذه البرمجيات لتعمل ضمن البيئة الجديدة.

من أكثر الأسئلة التي تطرح عند البدء بإستخدام لينكس: ما هي أفضل توزيعة للينكس؟ وأية توزيعة يجب أن أستخدم؟
في الواقع، لا توجد إجابة محددة لهذين السؤالين، ولا يوجد تفضيل مطلق لأي توزيعة على غيرها، فكل توزيعة تتمتع بخصائص تميزها، وإختيار التوزيعة التي ستستخدمها يعتمد على إختيار التوزيعة الأكثر ملاءمةً لمتطلباتك الخاصة. فيما يلي لمحة عن الخصائص الأساسية لأشهر توزيعات لينكس:

Mandrake: تعتمد هذه التوزيعة على توزيعة RedHat مع بعض التعديلات لغاية سهولة الإستخدام، وهي ملائمة جداً للإستخدامات المكتبية.

Debian: تعتبر أكثر التوزيعات تطوراً من الناحية التقنية. مجانية بالكامل وتعتمد بشكل مطلق على المعايير القياسية، إلا أن تثبيتها وإدارتها أصعب قليلاً من غيرها.

RedHat: أكثر التوزيعات شعبية على الإطلاق، إلا أن RedHat قد قامت مؤخراً بالتمييز بين إصداراتها الخاصة بالمخدمات وتلك المكتبية، وأسمت الأخيرة Fedora.

Slackware: والي كانت أولى توزيعات لينكس على الإطلاق، وتتمتع دوماً بالسبق في تضمينها أحدث إصدارات البرمجيات المرفقة. ولكنها ليست سهلة الإستخدام والإدارة.

هناك الكثير من التوزيعات الأخرى والتي لا يتسع المجال هنا لذكرها، وأود التنويه هنا إلى أن تجربة توزيعات لينكس المطورة محلياً جدير بالإهتمام كونها مصممة خصيصاً للإحتياجات الخاصة لبلدك ولغتك.

مزايا لينكس

لا يوجد أي نظام تشغيل يدّعي بأنه "كامل الأوصاف"، ولن يشذّ لينكس عن هذه القاعدة، ولكن ومع مرور الوقت فإن لينكس يدنو شيئاً فشيئاُ من الكمال. لقد أصبح لينكس أكثر سهولةً في التثبيت والإدارة، ولن تحتاج إلى إعادة تشغيل نظامك في كل مرة تجري فيها بعض التعديلات عليه. ولن أطيل عليكم بذكر كل ما أعتبره ميزةً من مزايا لينكس، بل سأحاول فيما يلي تلخيص أهم هذه المزايا:

الأمن

سيريحك لينكس من عناء برامج مكافحة الفيروسات وتطويراتها التي لا تنتهي، فعلى الرغم من أنه غير منيع ضد الفيروسات مئة بالمئة، ولكنه أقرب أنظمة التشغيل إلى هذه النسبة. على أرض الواقع، لا يوجد حتى هذا اليوم ما يسمى "فيروسات لينكس"، وهذا لا يعني أن

أحداً لم يحاول تطوير هذه الفيروسات، ولكنّ تصميم البنية الأساسية للينكس أولى اهمية قصوى للجانب الأمني. هذا يعني أن الفيروسات غير موجودة في عالم لينكس، ويتم التعامل مع جميع المشكلات الأمنية في لينكس بسرعة وكفاءة عبر مجموعات التطوير. يتم عادةً إصلاح الثغرات الأمنية في لينكس بعد عدة ساعات فقط من إكتشافها، ففي حال واجهت مشكلةً أمنيةً في نظامك، لن تعود مضطراً لإنتظار الإصدارة الجديدة من نظام التشغيل لحل هذه المشكلة.

الوثوقية

يتميز لينكس على وجه الخصوص ويتمتع بالقدر الأعلى من شعبيته في عالم التطبيقات الفائقة الأهمية، تلك التطبيقات التي تتطلب الكثير من الوثوقية و الأمن – كتطبيقات الإنترنت و التجارة الإلكترونية.

في عالم يكثر فيه حديث مستخدمي الحواسيب عن إعادة تشغيل حواسيبهم عدة مرات في اليوم، يتحدث مستخدمي لينكس عن أسابيع وشهور ظل خلالها حاسوبهم يعمل دون أن يضطروا لإعادة تشغيله.

من الطبيعي أن تتوقف بعض البرمجيات عن العمل لسبب ما أو لآخر، ولكنها لن توقف نظامك بأكمله عن العمل أيضاً.

الأداء

عدا عن كون لينكس نظام تشغيل متعدد المهام (أي أنه يستطيع القيام بأكثر من مهمة في آن واحد) فإنه أيضاً مبني أساساً ليكون متعدد المستخدمين، و هذا يعني أنه في مكتب ما على سبيل المثال بإمكان مخدّم واحد القيام بعدة مهام بنفس الوقت: يستطيع المدير المالي للشركة مثلاً استعراض حسابات الشركة بينما يقوم أحد الموظفين بلعب الشطرنج في غرفة أخرى.

أما في المنزل فإن تعدد المستخدمين يعني أن كلّ فرد من أفراد العائلة سيملك إعداداته الخاصة – كالتطبيقات البرمجية و شكل سطح المكتب ... الخ – دون التأثير على الإعدادات الخاصة لبقية أفراد الأسرة.

التحرر من القيود القانونية

عند إستخدامك لينكس ستتخلص من الحاجة إلى متابعة تراخيص إستخدام البرمجيات وأنظمة التشغيل، لأن إتفاقية ترخيص المصادر المفتوحة تتيح لك نسخ وإعادة توزيع البرمجيات مفتوحة المصدر دون أي خرق لقوانين حماية الملكية الفكرية.
ولكن عليك الإنتباه أن البرمجيات التي تعمل ضمن لينكس ليست جميعها مفتوحة المصدر، وبالتالي عليك التمعن في إتفاقيات الترخيص المرافقة لكل من هذه البرمجيات قبل إستخدامها للتأكد من ملاءمتها لإحتياجاتك.

عيوب لينكس

إن التوجّه نحو لينكس ينطوي على الكثير من المكاسب والمزايا، ولكن وللأمانة لا بد من ذكر نقاط الضعف التي يعاني منها لينكس وأخذها بعين الإعتبار عند إتخاذ أية قرارات تتعلق بالتحول إلى لينكس (والذي سنسهب في شرحه في فصول قادمة):

دعم التجهيزات

إذا أخذنا بعين الإعتبار الأطياف الواسعة من التجهيزات التي يدعمها لينكس سنجد دعم لينكس للتجهيزات والوحدات الطرفية متفوقاً على معظم أنظمة التشغيل الأخرى. ولكن

ولسوء الحظ مازال هناك بعض التجهيزات المصممة خصيصاً لنظام التشغيل ويندوز مما يتسبب في محدودية الدعم المتوفر لها ضمن لينكس (كبعض الماسحات الضوئية والطابعات).

من جهة أخرى، وبينما ستجد نفسك مضطراً لتحميل برامج القيادة لتشغيل التجهيزات مع أنظمة التشغيل الأخرى، يقوم لينكس باكتشاف وتمييز عدد هائل من التجهيزات والوحدات الطرفية تلقائياً دون أي تدخل من قبل المستخدم أو الحاجة إلى برامج القيادة لهذه التجهيزات.

يتميز مجتمع مستخدمي لينكس بديناميكية منقطعة النظير، فإذا وقعت عيناك على وحدة طرفية مثلاً فإنك تستطيع الجزم بأن أحد مطوري لينكس في مكان ما من هذا العالم قد اكتشفها أيضاً، وبالتالي لن يمر الكثير من الوقت قبل أن يصبح دعم وحدتك الطرفية جزءاً أساسياً من لينكس.

البرمجيات

هناك أعداد هائلة من التطبيقات التي تعمل ضمن لينكس، ولكن معظمها غير تجاري، وهذا يعني أن بإمكانك الحصول مجاناً على آلاف الألعاب والأدوات البرمجية والتطبيقات المكتبية لتشغيلها ضمن جهازك.

ولكن من جهة أخرى فإن معظم التطبيقات البرمجية التجارية (بما فيها أحدث إصدارات الألعاب والموسوعات) لا تعمل ضمن لينكس. إلا أن تزايد شعبية لينكس بشكل مطّرد سيغير ذلك قريباً.

خطوة نحو المجهول

علينا مواجهة الواقع بتجرد: فالبعض يعتبر أن الحول نحو لينكس خطوةً نحو المجهول، فالأمور لن تبقى على ما اعتادوا عليه في أنظمة التشغيل التي تعاملوا معها. كل ما يتطلبه التغلب على هذا الحاجز النفسي هو القليل من التدريب على النظام الجديد.

مع ذلك، فإن تحولك إلى لينكس إذا كنت معتاداً على إستخدام الواجهات الرسومية لويندوز وإستخدام التطبيقات المكتبية العادية كبرامج البريد الإلكتروني ومحررات النصوص لن يسبب لك الكثير من العناء. لقد تطورت الواجهات الرسومية في لينكس بشكل كبير جداً، وهي فائقة السهولة في الإستخدام، عدا عن أنك تستطيع إستخدام معظم مهاراتك التي اكتسبتها من عملك مع أنظمة التشغيل الأخرى.

من يستخدم لينكس؟

ينتشر لينكس بكثرة في أوساط المبرمجين وشركات الإتصالات ومحرري الوسائط المتعددة (والأمثلة كثيرة) نظراً لما يقدمه من خدمات لتلبية متطلباتهم المختلفة.

يجد المبرمجون ضالتهم في لينكس لإنخفاض تكاليفه بالدرجة الأولى، فلينكس يزودهم ببيئة تطوير متكاملة لا تتطلب تجهيزات متخصصة. وبما أن لينكس متوافق تماماً مع معايير يونيكس فإن أي برنامج سيتم تطويره ضمن لينكس سيعمل مباشرة ضمن أية إصدارة أخرى من يونيكس.

ويتميز لينكس على وجه الخصوص في عالم الشبكات الحاسوبية، وقد استخدم على نطاق واسع لتوفير الخدمات التشبيك في الكثير من شركات الإتصالات ومزودي خدمات الإنترنت.

ويضمن الدعم الكامل لعدد كبير من بروتوكولات الإتصالات (والتي تستخدم نفسها في المخدمات) ضمن لينكس التفاعل مع الشبكات الحاسوبية للشركات والمؤسسات بسهولة ويسر، من اليسير على سبيل المثال مشاركة الملفات ضمن الشبكة أو حتى تشغيل البرمجيات على حواسيب أخرى. ولا يقف تناغم لينكس المميز مع الشبكات عند هذا الحد، بل يستطيع لينكس حتى التحدث بلغة أنظمة التشغيل الأخرى (مثل مايكروسوفت ويندوز) ليندمج معها في شبكة واحدة ويقدّم خدمات مشاركة الملفات والطباعة لحواسيب ويندوز المربوطة على الشبكة –ويدعي الكثيرون أن لينكس يندمج مع شبكات ويندوز بسرعة أكبر من ويندوز نفسه!

أما مطورو أنظمة التشغيل فقد كانوا السبّاقين لإستخدام لينكس، فقد اعتبروه ملعباً رحباً لإبراز مواهبهم وتطويرها، فقاموا بالعمل الدؤوب ليل نهار على تطوير وتحسين لينكس حتى أصبح قادراً على منافسة أكثر أنظمة التشغيل عتواً.

وفي عالم الوسائط المتعددة، لم يكن نجاح لينكس إستثناءاً. فقد نجح لينكس في خطف الأضواء لما تميّز به من دعم لطيف واسع من تجهيزات الوسائط المتعددة كبطاقات الإظهار والصوت. وتم تعجيل العديد من بيئات تطوير الوسائط المتعددة لتعمل ضمن لينكس (مثل MESA 3D). ويعتبر برنامج تحرير الصور GIMP (وهو المكافئ المجاني لبرنامج Adobe Photoshop ضمن لينكس) أداةً لا غنى عنها للكثير من الفنانين. ولنذكر مثالاً على استخدام لينكس في عالم الوسائط المتعددة وإذا كنت من هوّاة الأفلام العاطفية وشاهدت المؤثرات المرئية في فيلم (تايتانيك Titanic) فلا شكّ أنّك ستقدّر إمكانيات لينكس الذي استخدم في إنتاجها.

وقد استخدم لينكس بكثرة في التطبيقات الضخمة كأنظمة الإتصالات وأنظمة المعلومات الجغرافية وأبحاث الفضاء، كما تستخدم عدّة مشاف لينكس لإدارة سجلات المرضى والأطباء، وهو ما يثبت تماثل وثوقية لينكس للإصدارات الأخرى من يونيكس.

يتوسع استخدام لينكس وإنتشاره في مجالات شتى، ومع تطور الواجهات الرسومية للينكس وازدياد سهولة إدارته سيجد طريقه بسرعة إلى عالم الحواسيب الشخصية المستخدمة في المنازل والمكاتب، مما سيوفر على المستخدمين الكثير من المال ويزودهم بأنظمة أكثر مرونةً وقابليةً للتعديل والتخصيص.

من أين أستطيع الحصول على لينكس؟

هناك الكثير من طرائق الحصول على لينكس، فبعد أن تقرر التوزيعة التي ستستخدمها عليك الحصول على هذه التوزيعة، إما عبر الإنترنت أو بشرائها على قرص مدمج (تستطيع أيضاً إستعارة نسخة صديق لك ونسخها دون أن تخرق أياً من قوانين حماية حقوق الملكية الفكرية). قد تتضمن بعض التوزيعات برمجيات غير مفتوحة المصدر وبالتالي عليك الإنتباه لمراعاة إتفاقيات ترخيص هذه البرمجيات (أي أنك قد لا تتمكن من نسخ هذه البرمجيات). على أية حال فإن توزيعات كل منRedHat و Mandrake و Debian و Slackware (وغيرها) ملتزمة بحرّية المستخدم ولا تتضمن أية برمجيات غير مفتوحة المصدر، وبإمكانك نسخ هذه التوزيعات كما تشاء.

لا بد من التذكير هنا بأن إتفاقية ترخيص GPL لا تعني أن البرمجيات الحرّة يجب أن تكون مجّانية، فهي تتيح لك الحصول على مقابل مادّي لقاء توزيع وتثبيت وتوفير الخدمات للبرمجيات الحرّة.

يحتوي موقع Metalab على العديد من توزيعات لينكس التي تستطيع الحصول عليها مجاناً من الموقع مباشرة: ftp://metalab.unc.edu/pub/Linux/distributions

سيستغرق حصولك على لينكس عبر الإنترنت الكثير من الوقت، خصوصاً إذا كانت وصلتك إلى الإنترنت بطيئة، لذلك قد يكون من الأفضل أن تبحث عن مصدر محلّي يقوم بتوزيع لينكس. تأكد دائماً من حصولك على أحدث إصدار من التوزيعة التي تريدها، ولا تقم بأي حال بتثبيت توزيعة تم تطويرها وتوزيع إصدار جديد منها.

الدعم الفني للينكس

من أكثر الأسئلة شيوعاً عند التفكير بالتحول إلى لينكس وأنظمة المصادر المفتوحة: عندما أشتري برنامجاً مغلقاً من شركة ما فإن هذه الشركة توفر لي الدعم الفني لهذا البرنامج، من سيقدّم لي الدعم الفني للينكس وبرمجيات المصادر المفتوحة؟

تتعلق الإجابة على هذا السؤال بشكل مباشر بفلسفة لينكس والبرمجيات مفتوحة المصدر، ففي عالم لينكس يسود إحساس عميق بالإنتماء إلى المجموعة بين مستخدمي لينكس، هذا الإحساس يدفع كلاً منهم للمساهمة في تقديم الدعم والمساندة للمجموعة، في حين يتردد مستخدموا البرمجيات المغلقة في تقديم الدعم والمعلومات للآخرين إنطلاقاً من أنهم غير مدينين لهذه البرمجيات أو مطوريها، فقد دفعوا ثمنها!

أما مستخدموا لينكس فتجدهم متساندين إلى أبعد الحدود، يساعد بعضهم بعضاً ضمن آلية عمل جماعية مميزة وفائقة الفعالية، ويستطيع المستخدمون الحصول على الدعم الفني بمستويات تفوق بكثير تلك التي قد يحصلون عليها من منتجي البرمجيات المغلقة. تترافق معظم البرمجيات مفتوحة المصدر مع قوائم تراسل إلكتروني يمكن من خلالها طلب الدعم الفني من مطوري هذه البرمجيات مباشرةً. كما يوجد العديد من قوائم التراسل الموزّعة جغرافياً في جميع أنحاء العالم حيث تتم إجابة معظم الأسئلة المطروحة خلال عدّة ساعات. وبعد فترة إستكشاف قصيرة لهذه الموارد يجد مستخدموا لينكس الجدد أنفسهم غارقين في جوّ ممتع مميز من الحوارات عن أية مشكلة حاسوبية قد تواجههم.

وهناك أيضاً المجموعات الإخبارية والتي تحتوي على كنوز من المعلومات الفنيّة القيّمة نتيجة نقاشات وحوارات المستخدمين والمطورين والتي تخدم المبتدئين والخبراء على حدّ سواء. تتميز المجموعات الإخبارية بتواصلها مع أكبر عدد ممكن من المستخدمين.

كما تشكّل الإنترنت مصدراً فائق الأهمية للمعلومات والدعم الفنّي، ونتيجةً للتفاعل المستمر بين مستخدمي لينكس لمناقشة المشاكل والأمور الفنيّة فإن 99% من المشاكل التي قد تعترض مستخدمي لينكس قد تم طرحها ومعالجتها ونشرها للجميع على الإنترنت، مما سيلغي الحاجة لطلب المساعدة من أي كان.

وأخيراً، تقدم عدة شركات دعماً تجارياً للينكس لقاء مقابل مادّي، وفي حال كان موضوع الدعم الفني يشكل عائقاً يحول دونك ودون تحولك إلى لينكس فإن هذه الشركات ستقدّم لك الضمانة التي ستريح بالك.

أسئلة عليك إجابتها

قبل إتخاذك أي قرار يتعلق بخيارك الإستراتيجي لنظام التشغيل لينكس لا بد لك من طرح الأسئلة التالية ومحاولة البحث عن إجابات تدعم قراراتك:

ما هي الكلفة الكلية لاستخدام لينكس مقارنةً بأنظمة التشغيل المغلقة؟

على الرغم من كون لينكس مجانياً فإن إعداد نظام التشغيل قد يتطلب بعض الخبرات والمهارات، خصوصاً في أنظمة الشركات والمؤسسات، وقد ينطوي ذلك على بعض التكاليف في الوقت أو نفقات الدعم والإستشارات، أما الأنظمة المغلقة فتتطلب دفع رسوم رخصة الإستخدام.

تغفل معظم النقاشات حول تكاليف أنظمة التشغيل (ونحن هنا نتحدّث بشكل رئيسي عن مخدّمات الشركات والمؤسسات) النفقات المترتبة خلال دورة الإستثمار الفعلية للمخدّم مما يؤدي إلى إستنتاجات خاطئة مفادها أن إستخدام لينكس لا يكلّف شيئاً أو بالعكس، لا ينصح باستخدامه نظراً لارتفاع تكاليفه!

لا تقارب أي من هاتين الحالتين المتناقضتين الواقع الفعلي، فالتكاليف الفعليّة الكليّة لمخدّم ما تشمل:

- ثمن رخصة إستخدام نظام التشغيل.
- تكاليف تراخيص إستخدام البرمجيات الإضافية الضرورية.

- تكاليف التجهيزات.
- توافر التجهيزات المتوافقة للعمل مع نظام التشغيل المستخدم.
- تكاليف الإعداد.
- تكاليف الدعم الفنّي.
- التكاليف الناجمة عن الخسائر التي قد تحصل أثناء توقّف المخدّم عن العمل.
- التكاليف الناجمة عن الخسائر التي قد تسببها الثغرات الأمنية.
- تكاليف الصيانة.
- تكاليف التطويرات الأساسية.
- التكاليف الناجمة عن ضرورة إستخدام مخدّمات متعددة (يستطيع لينكس على سبيل المثال العمل كمخدّم للبريد الإلكتروني والملفات والوب في آن واحد مستخدماً جهازاً واحداً فقط مما يعني توفيراً هائلاً في التكاليف).

عند أخذ جميع العوامل السابقة بعين الإعتبار تبدو بوضوح النفقات التي ستوفرها أية شركة أو مؤسسة عند إختيارها مخدّماً يستخدم لينكس وبرمجيات المصادر المفتوحة.

ومقارنةً مع أنظة يونيكس الأخرى، يعتبر لينكس أسهل هذه الأنظمة استخداماً على الإطلاق نظراً لاستخدامه الواسع وسهولة الحصول على المعلومات والدعم الفنيّ له، ويتمتع لينكس بمشروع توثيق ضخم يتولى تنظيم معلومات لينكس بصيغة سهلة وميسرة لجميع المستخدمين.

تتطلب بعض أنظمة يونيكس تجهيزات خاصّة لتعمل ضمنها، مما يجعل تكاليفها باهظة جداً، وخاصةً عندما يتطلب الأمر دفع رسوم تراخيص استخدام أنظمة التشغيل أيضاً.

كيف يقارن أداء لينكس بأنظمة التشغيل الأخرى؟

يقوم لينكس بأداء نفس المهام التي يقوم بها أي نظام تشغيل آخر بشكل أفضل من 50-100% مستخدماً نفس التجهيزات.
لقد أجريت العديد من المقارنات المضللة بغاية إظهار لينكس بشكل أفضل أو أدنى من الأنظمة الأخرى، معظم هذه المقارنات كانت تجرى في ظروف خاصة تشوّه مصداقية نتائجها، ولكن في بعض المقارنات التي تعتبر أكثر حياديةً وموضوعيةً، تفوّق لينكس على ويندوز في تخديم الوب، الملفات أو قواعد البيانات.

يظهر فرق الأداء عند استخدام لينكس بوضوح للمستخدمين ومدراء الأنظمة، كسرعة التعامل مع الملفات وعدم تراجع الأداء عند تشغيل عدّة خدمات في آن واحد.

ألا يشكل إعتماد لينكس على بنية يونيكس خياراً عفى عليه الزمن؟

إن المبادئ الأساسية التي تحدد بنية أنظمة التشغيل لم تتغير منذ ابتكار فكرة أنظمة التشغيل (منذ ما يزيد على 40 عاماً) والتي تطور في مراكز البحوث والجامعات التي تضع المفاهيم النظرية لعلوم الحاسوب، أما الشركات فتقوم فقط بتطبيق هذه المفاهيم.

هناك الكثير من المفاهيم النظرية لأنظمة التشغيل والتي تتفاوت في التعقيد ومجالات التطبيق، وتتميز بنية يونيكس بين هذه المفاهيم لمرونتها وأمنها وقابليتها للتعديل.

ما هي الأمور الأساسية الواجب أخذها بعين الإعتبار عند التحول إلى لينكس؟

سنقوم بالإجابة عن هذا السؤال بالتفصيل في الفصل (التحول إلى لينكس)

الفصل الثالث

المصادر المفتوحة واللغة العربيّة

مقدّمة

إن استخدام برمجيات المصادر المفتوحة في الوطن العربي واعتمادها على نطاقٍ واسع مرتبط بشكل أساسي بقدرة هذه البرمجيات على التعامل مع اللغة العربيّة، وبدون دعم اللغة العربيّة ستبقى هذه البرمجيات عاجزةً عن تبوؤ مكانتها الحقيقية في الوطن العربي ولن يكون المستخدم العربي قادراً على الإفادة من مزايا هذه البرمجيات وسيبقى مضطراً إلى الإعتماد على الشركات المنتجة للبرمجيات المغلقة لتوفير الحلول التقنية التي يتطلبها الإنتقال إلى مجتمع المعلومات.

لذلك لا بدّ من العمل بجدّ على توفير الدعم الكامل للغة العربيّة في هذه البرمجيات وذلك لكي لا يبقى الوطن العربي مستثنىً من ثمار ثورة المصادر المفتوحة.

خصوصية اللغة العربيّة

تتمتع اللغة العربيّة بمجموعة من الخصوصيات التي تميّزها عن غيرها من اللغات، وتتطلب هذه الخصوصيات معاملةّ خاصّة أثناء إضافة دعم اللغة العربيّة إلى برمجيات المصادر المفتوحة، وهو ما تسبب في تأخّر إضافة دعم اللغة العربيّة إلى هذه البرمجيات نسبياً، حيث أن التعامل مع خصوصيات اللغة العربيّة يتطلب تدخّل مطورين ملمين بهذه الخصوصيات

وأساليب التعامل معها، وهو ما أدّى بدوره إلى تأخّر دخول مفاهيم البرمجيات مفتوحة المصدر إلى الوطن العربي عن غيره من دول العالم.

تتجلى خصوصيات اللغة العربيّة فيما يتعلق بتعريب البرمجيات بما يلي:

- دعم أطقم المحارف العربيّة
- الخطوط العربية
- تحليل السياق
- دعم ثنائية الإتجاه
- واجهة التطبيقات
- التشكيل والرموز الخاصّة

دعم أطقم المحارف العربية

طاقم المحارف هو الخريطة التي يقابل فيها الحاسوب بين كل حرف في لغةٍ ما (وهو ما يستطيع الإنسان التعامل معه ومعالجته) مع رمزٍ ثنائيٍ يستطيع الحاسوب التعامل معه ومعالجته.

هناك الكثير من أطقم المحارف المتوفرّة لكل لغة من لغات العالم، بل يوجد حتى عدّة أطقم محارف للغة الواحدة، حيث قامت العديد من الجهات التي طوّرت أنظمة التشغيل والبرمجيات التطبيقية كالشركات ومركز البحوث بتطوير أطقم محارف خاصّة تمكّنها من إضافة دعم اللغات المختلفة إلى برمجياتها. وقد أدّى ذلك إلى تواجد مجموعةٍ من أطقم المحارف غير المتوافقة مع بعضها البعض لكل لغة، فنظام التشغيل المتضمن طاقم محارف

اللغة العربيّة Windows-1256 مثلاً لن يستطيع التعامل مع الملفات والبيانات المخزّنة باستخدام طاقم المحارف Unicode. هذا الإختلاف بدوره أدى إلى عزل الأنظمة والبرمجيات المختلفة عن بعضها البعض.

لقد تم تطوير بعض البرمجيات التي تقوم بالتحويل من طاقم محارف إلى آخر لحلّ هذه المشكلة الناتجة عن عدم توافق أطقم المحارف مع بعضها البعض، ولكنّ هذا الحل يصطدم بعوائق عدّة عند تطبيقه عملياً. ويشكّل اهتمام الشركة صاحبة المنتج البرمجي بالسوق العربية المحرّك الأول والأخير لدفعها لتوفير دعم اللغة العربيّة في برمجياتها، وبما أنّ معظم البرمجيات التجارية مغلقة ولا يمكن الوصول إلى شيفرتها المصدرية فقد كان مصير أية مبادرة لحل مشاكل اللغة العربيّة في البرمجيات المغلقة الفشل منذ البداية.

يختلف الموضوع كليّةً في عالم المصادر المفتوحة، حيث يمكن الحصول على الشيفرة المصدرية وتعديلها بما يتلاءم مع احتياجات اللغة العربيّة أو غيرها. وتؤدي آليّة مشاركة الأفكار والخبرات إلى تسريع العم وضمان مستوىً عالٍ جداً من النوعية في البرمجيات المطورة.

إن وجود طاقم محارف معياري موحّد للغة العربية سيسهل بشكل كبير من مهمّة تعريب البرمجيات مفتوحة المصدر، وسيتجاوز العقبات التي نشأت نتيجة عدم توافق أطقم المحارف العربية المختلفة، كما سيتيح بناء قواعد البيانات العربية في جميع أنحاء العالم بصيغة قياسية تتيح سهولة تبادل هذه البيانات ومعالجتها حتى بين الأنظمة المختلفة.

لقد تمّ تطوير طاقم محارفٍ موحد يدعى Unicode، وهو يتجاوز متطلبات توحيد أطقم محارف اللغة العربية كونه عبارةً عن طاقم محارفٍ موحد لجميع لغات العالم، أي أنّه قادر على التعامل مع أي حرفٍ في أي لغةٍ من لغات العالم، ويعتبر هذا الإنجاز من أهم الإنجازات في صناعة تقنية المعلومات لما يوفره من سهلة وسرعة في تطوير البرمجيات متعددة اللغات.

إن بناء البرمجيات التطبيقية بإعتماد معايير اليونيكود يعني أنها ستحتوي تلقائياً على دعم اللغة العربيّة (شأنها شأن أية لغة أخرى)، وهو ما يجري حالياً في عالم المصادر المفتوحة، حيث يتم تطوير الغالبية العظمى من البرمجيات بناءً على معايير اليونيكود،. لا بدّ على أية حالٍ من التنويه إلى ضرورة إضافة دعم أطقم المحارف الأخرى في هذه البرمجيات عبر أدوات التحويل المذكورة آنفاً وذلك لتجاوز العقبات التي قد تظهر عند الضرورة للتعامل مع بياناتٍ مرمّزةٍ باستخدام أطقم المحارف الأخرى.

الخطوط العربيّة

يتطلب التعامل مع اللغة العربيّة توفّر مجموعةٍ من الخطوط العربية ضمن نظام التشغيل أو البرنامج. والخطوط، مثلها كمثل أي برنامج أو عملٍ فني، لها حقوق ملكية محميّة، أي أنّ استخدام هذه الخطوط دون إذن مسبقٍ من صاحبها يعتبر خرقاً لقوانين ومعاهدات حماية الملكية الفكرية ويعاقب عليه القانون.
نظراً لطبيعة برمجيات المصادر المفتوحة فإن أية خطوطٍ ستتضمنها يجب أن تكون حرةً من أية قيود تفرضها قوانين حماية الملكية الفكرية، لأن هذه البرمجيات (وبالتالي الخطوط)

ستوزّع ضمن ترخيص المصادر المفتوحة والذي يمنح المستخدم الحقّ في تعديل وإعادة توزيع هذه البرمجيات أو أي جزءً منها دون الحصول على موافقةٍ مسبقةٍ من مطورها.

بما أن دعم اللغة العربيّة المتوفر حالياً في البرمجيات المغلقة كان نتاج عمل شركات تطوير البرمجيات التجارية فإن معظم خطوط اللغة العربية المتوفرة حالياً تنطوي ضمن إطار حقوق الملكية الفكرية لهذه الشركات ولا يمكن استخدامها أو تعديلها دون الحصول على إذنٍ من هذه الشركات، وهو ما قد يشكّل عائقاً في توفير دعم اللغة العربيّة في برمجيات المصادر المفتوحة.

ومن هنا تبرز أهميّة تطوير خطوط عربية (مفتوحة المصدر)، أي أنّها متاحة لأي كان لاستخدامها وتعديلها وإعادة توزيعها تماماً مثل برمجيات المصادر المفتوحة. وقد تمّ بالفعل تطوير عدّة مجموعات من الخطوط مفتوحة المصدر والتي تستخدم حالياّ في البرمجيات مفتوحة المصدر المعرّبة، ولكننا نطمح نحو توفير عددً أكبر من هذه الخطوط لمنح المستخدم المزيد من المرونة والخيارات.

تحليل السياق

يتغير شكل الحرف في اللغة العربيّة تبعاً لموقعه من الكلمة، أي أن شكل حرف الباء مثلاّ عندما يقع في بداية الكلمة يختلف عن شكله عندما يقع في وسط الكلمة أو آخرها. ومع أن الحرف هو نفسه بالنسبة للحاسوب، إلا أن (شكل) هذا الحرف يتغير تبعاً لموقعه من الكلمة. يسمى شكل الحرف الذي يظهر على الشاشة أو الطابعة Glyph.

إن إظهار الكلمات العربيّة بشكل صحيح يتطلب إظهار كل حرف بشكله الصحيح، وهو ما يتطلّب معرفة موقع الحرف في الكلمة لإظهار الشكل الصحيح الموافق. هذه العملية تدعى (تحليل السياق) حيث يقوم الحاسوب بمتابعة موقع الحرف في الكلمة وإظهار الشكل الصحيح الموافق لهذا الموقع.

ولكي يحتوي أي برنامجٍ على دعم اللغة العربيّة لا بد أن تتوافر فيه إمكانية تحليل السياق ليكون قادراً على إظهار الأحرف والكلمات العربية بشكل صحيح.

يمكن إضافة تحليل السياق إلى كل برنامج بشكل مستقل، كما يمكن إضافته إلى نظام التشغيل نفسه، وعندها يمكن استخدام آليات تحليل السياق الموجودة ضمن نظام التشغيل من قبل التطبيقات البرمجية لتتمكن من توفير دعم اللغة العربيّة.

في بداية مبادرات تطوير دعم اللغة العربيّة في البرمجيات مفتوحة المصدر قامت مجموعات المطورين بإضافة آليات تحليل السياق إلى البرمجيات التي يقومون بتطويرها كل على حدة، ولكن هذا الأسلوب ينطوي على بعض المشاكل، والتي قد تتجلّى في عدم توافق آليات تحليل السياق بين البرمجيات المختلفة أو بين البرمجيات ونظام التشغيل الذي تعمل ضمنه، عدا عن صعوبة بناء وإضافة آليات تحليل السياق بشكل منفصل لكل من البرمجيات المتوفرّة في عالم المصادر المفتوحة. ولذلك فقد برزت الحاجة إلى وجود آليةٍ معياريةٍ لتحليل السياق تستخدم لإضافة دعم اللغة العربية في البرمجيات مفتوحة المصدر.

إن إضافة آليات تحليل السياق إلى بيئات التطوير المستخدمة في تطوير برمجيات المصادر المفتوحة سيؤدّي إلى توفير تحليل السياق في جميع البرمجيات التي ستبنى باستخدام هذه

البيئات، ليختصر بذلك الوقت والجهد اللازمين للتعامل مع كل برنامجٍ على حدة، وليضمن توافق هذه الآليات ضمن جميع هذه البرمجيات. وسيجعل ذلك إضافة دعم اللغة العربية إلى برمجيات المصادر المفتوحة أمراً اكثر سهولةً وسرعةً، وسيؤدي إلى توفير النسخ المعرّبة من هذه البرمجيات في نفس توقيت إصدار النسخ الأساسية باللغات الأخرى.

وبالفعل فقد تمت إضافة آليات تحليل السياق إلى بيئات تطوير البرمجيات مفتوحة المصدر والتي تعتمد اليونيكود، ومنها علي سبيل المثال بيئة Qt المستخدمة في تطوير بيئة سطح المكتب KDE وبيئة +GTK المستخدمة في تطوير بيئة سطح المكتب GNOME.

وبمجرد استخدام أي من هذه البيئات لتطوير البرمجيات مفتوحة المصدر فإن ذلك سيضمن توفّر دعم تحليل السياق في هذه البرمجيات.

دعم ثنائية الإتجاه

تكتب اللغة العربيّة من اليمين إلى اليسار، ولكن وفي بعض الحالات قد يحتوي النص العربي على معلومات تكتب بالإتجاه المعاكس (من اليسار إلى اليمين) كالأعداد والنصوص اللاتينية، ولا بدّ لأي برنامج يدعم اللغة العربيّة أن يوفرّ إمكانية تغيير إتجاه الكتابة ضمن النص عند الحاجة، وهو ما يدعى بدعم ثنائية الإتجاه.

تخيّل مثلاً أنّك تقوم بطباعة نصٍ باللغة العربيّة (من اليمين إلى اليسار) وأردت إدراج كلمةٍ لاتينيةٍ ضمن هذا النص، فبدون توفّر دعم ثنائية الإتجاه في البرنامج الذي تستخدمه سيظهر

ترتيب الأحرف اللاتينية كما في الأحرف العربيّة من اليمين إلى اليسار وستظهر الكلمة اللاتينية بالمقلوب.

لقد تم تطوير عدّة خوارزميات لدعم ثنائية الإتجاه ضمن التطبيقات البرمجية، وهي متوفرّة ضمن ترخيص المصادر المفتوحة، منها Pretty Good BiDI و اليونيكود. كما تقوم شركات تطوير البرمجيات التجارية بتطوير الخوارزميات الخاصّة بها.

إن إضافة هذه الخوارزميات إلى بيئات تطوير البرمجيات مفتوحة المصدر سيوفّر دعم ثنائية الإتجاه ضمن البرمجيات المطورة باستخدام هذه البيئات دون الحاجة إلى تعديل كل برنامج على حدة والوقوع من جديد في مصيدة عدم توافق الخوارزميات المستخدمة في البرمجيات المختلفة.

قامت بعض مجموعات تطوير البرمجيات مفتوحة المصدر ببناء وإضافة خوارزميات دعم ثنائية الإتجاه إلى البرمجيات التي تقوم بتطويرها بشكل مستقل، وعدا عن مشاكل عدم التوافق فإن هذا المهمة تستغرق الكثير من الوقت لتعديل الشيفرات المصدرية للبرمجيات وتجربتها، وهو ما أدّى إلى تأخر ظهور النسخ المعرّبة من البرمجيات مفتوحة المصدر.

تحتوي بيئات التطوير المبينة باستخدام معايير اليونيكود على دعم ثنائية الإتجاه، وبالتالي فإن أية برمجيات سيتم تطويرها باستخدام هذه البيئات ستحتوي تلقائياً على دعم ثنائية الإتجاه المطلوب توافره لدعم اللغة العربية في هذه البرمجيات.

واجهة التطبيقات

لأسبابٍ تجارية، قامت بعض الشركات المطورة للبرمجيات التجارية بتسويق مفهومين متعلقين بدعم اللغة العربيّة في البرمجيات، وقد شاع استخدام هذين المفهومين بشكل كبير. حيث يدعى البرنامج أو نظام التشغيل الذي يستطيع التعامل مع اللغة العربيّة ولكن واجهاته التي يتفاعل المستخدم معه من خلالها غير معرّبة بالبرنامج الذي (يدعم اللغة العربية)، أما البرنامج الذي يحتوي على دعمٍ كاملٍ للغة العربيّة وتكون واجهته متوفّرة باللغة العربيّة أيضاً بالبرنامج (المعرّب).

إن استخدام هذين المفهومين لتصنيف البرمجيات التي تدعم اللغة العربيّة ينطوي على الكثير من عدم الدقّة، فلا يجوز اعتبار البرنامج معرّباً ما لم يحتوي على واجهةٍ عربيّةٍ تتيح لأي مستخدمٍ التعامل مع هذا البرنامج حتى إذا كان غير متمكنٍ من لغات أخرى غير اللغة العربيّة. ولا بدّ لنا من تفهم هذا الموضوع بشكل جيّد لأن تطوير البرمجيات مفتوحة المصدر يهدف في الدرجة الأولى إلى توفير برمجيات ذات نوعيةٍ عاليةٍ تلبّي متطلبات مستخدميها وليس تحقيق المزيد من الأرباح.

قد يكون من المقبول عملياً التعامل مع البرمجيات مفتوحة المصدر التي توفّر دعم اللغة العربيّة دون وجود واجهات استخدام معرّبة لها بشكل مرحلي ريثما يتم تعريب هذه الواجهات، وذلك لتجنّب المزيد من التأخير في استخدام البرمجيات مفتوحة المصدر وللحصول على آراء المستخدمين لاستخدامها في تعريب واجهات التطبيقات.

ينطوي تعريب واجهة المستخدم على تعريب عدّة مكونات أساسية:

- واجهة المستخدم.

- رسائل الخطأ.
- معلومات المساعدة.

يعتبر تعريب واجهة المستخدم الخطوة الأولى، حيث يتم تعريب جميع أجزاء واجهة البرنامج التي يتعامل معها المستخدم للتفاعل مع البرنامج، كالأوامر والخيارات وغيرها.
وفي الخطوة التالية يتم تعريب رسائل الخطأ للبرنامج، وهي العبارات التي يعرضها البرنامج لإعلام المستخدم عند حدوث خطأٍ ما.
ومن ثمّ يتم تعريب ملفات المساعدة المتوفرة مع معظم البرمجيات والتي تقدّم للمستخدم معلوماتٍ تفصيليةً أو مختصرةً عن كيفية عمل البرنامج وطريقة التعامل معه.

لا يعتبر البرنامج معرّباً بالكامل ما لم يتم تعريب جميع المكونات المذكورة آنفاً من واجهة المستخدم، لأن ذلك سيحول دون استثمار هذا البرنامج بشكلٍ تام من قبل جميع المستخدمين العرب، وهو ما يتعارض مع تعريف البرمجيات مفتوحة المصدر والتي يجب أن تكون متاحةً لأكبر قدرٍ ممكن من المستخدمين. كما أنّ ذلك سيحول دون انتشار فلسفة المصادر المفتوحة بالشكل المطلوب في الوطن العربي.

إنّ تعريب واجهة المستخدم يعتبر من أبسط مراحل تعريب البرمجيات من الناحية التقنية، ولكنّه يستهلك أكبر قدر من الوقت والعمل. فالبرمجيات مفتوحة المصدر مبنية أساساً لتحتوي على دعم اللغات المتعددة، مما يجعل من عملية تعريب واجهة الإستخدام مجرّد ترجمةٍ للعبارات المستخدمة إلى اللغة العربيّة وتنسيق هذه العبارات المترجمة في ملفاتٍ خاصّةٍ.

من أهم المشكلات التي تواجه عملية تعريب البرمجيات مفتوحة المصدر هو عدم وجود مصدر واحد لترجمة كل المصطلحات المستخدمة أثناء عملية التعريب، وهيمنة المصطلحات المعرّبة من قبل بعض الشركات التجارية وانتشارها على نطاقٍ واسعٍ أضحت معه تعتبر التعريب الأساسي على الرغم من كل ما تحتويه من أخطاء.

يضاف إلى ذلك المشكلة الأساسية في عالم المصادر المفتوحة وهي غياب المركزية في تنسيق نشاطات التطوير، مما قد يتسبب في تكرار العمل نفسه من قبل أكثر من مجموعة. ولذلك تبرز هنا الحاجة إلى إنشاء جهةٍ خاصةٍ للتنسيق بين نشاطات التعريب المختلفة، وقد تقوم هذه الجهة بجمع المصطلحات المترجمة في معجمٍ معياريٍ واحد يستخدم في جميع نشاطات التعريب لضمان الحصول على تعريبٍ موّحدٍ لجميع البرمجيات مفتوحة المصدر مما سيسهّل بالتالي تعامل المستخدم العربي مع هذه البرمجيات.

إن عدد البرمجيات مفتوحة المصدر هائل، وهي في ازديادٍ مستمرٍ يوماً بعد يوم، ولا بدّ لنا من تكثيف جهودنا لتعريب أكبر قدرٍ ممكن من هذه البرمجيات للتسريع في إدخال المصادر المفتوحة إلى وطننا العربي والإفادة من كل ما توفّره من ميزات.

التشكيل والرموز الخاصّة

تحتوي جميع اللغات على بعض الرموز الخاصّة والتي تدعى Neutrals (كالأرقام والأقواس وعلامات التنصيص). ويجب التعامل مع هذه الرموز بشكل صحيح أثناء عمليّة تعريب البرمجيات ليكون هذا التعريب كاملاً.

تستخدم اللغة العربيّة حالياً الأرقام الهندية، في حين تستخدم اللغات اللاتينية الأرقام العربية، وفي كلتا الحالتين فإن الأعداد تكتب من اليسار إلى اليمين، وهو ما يمكن تجاوزه باستخدام دعم ثنائية الإتجاه. ولقد ظهرت عدّة أصواتٍ تنادي بالعودة إلى استخدام الأرقام العربية في الكتابات والنصوص العربيّة عوضاً عن الأرقام الهنديّة المستخدمة حالياً.

فيما يتعلق بتعريب البرمجيات فإن الأرقام تتطلب معالجةً خاصّةً، وذلك لكي يتمكّن الحاسوب من معالجة هذه الأرقام والتعامل معها. أي أنّ الأرقام يجب أن تظهر وتخزّن كأرقام وليس كأشكالٍ لهذه الأرقام فقط (بمعنى أن يكون الشكل هو شكل الرقم نفسه ولكن ما يتم تخزينه في المقابل ضمن الحاسوب هو هذا الشكل باعتباره حرفاً وليس الرقم نفسه وبالتالي لن يتمكن الحاسوب من إجراء أية عمليات حسابية على هذا الشكل المخزّن). ومن هنا أتت تسمية Neutrals لهذه الرموز والأرقام، وهي تعني أنّ وظيفة هذه الرموز لا تتغير من لغةٍ إلى أخرى، فالأرقام أرقام في جميع اللغات.

ومن خصوصيات للغة العربيّة أيضاً إحتوائها على التشكيل، حيث يتغير لفظ كلّ حرفٍ تبعاً للعلامة التي تضاف إليه (كالفتحة، الضمّة، الكسرة، الشدّة، السكون والتنوين). والتشكيل أساسي في اللغة العربيّة، ويعتبر جزءاً لا يتجزأ منها، ولا بدّ لتعريب أيٍ من البرمجيات أن تتمكّن هذه البرمجيات من التعامل مع التشكيل.

ويضاف إلى ذلك أسلوب تعامل البرمجيات المعرّبة مع علامات التنصيص (كالفاصلة وعلامة الإستفهام والنقطة)، والتي تعتبر أيضاً جزءاً أساسياً من عملية التعريب.

متطلبات أخرى

ينتشر في الوطن العربي استخدام التقويم الهجري بالإضافة إلى التقويم الميلادي، والذي يستخدم الأشهر الهجرية للتعبير عن التواريخ. ولذلك ينبغي أن تتم معالجة موضوع التعامل مع التواريخ الهجرية جنباً إلى جنب مع التواريخ الميلادية أثناء عملية تعريب البرمجيات مفتوحة المصدر. ويمكن إضافة دعم التقويم الهجري على مستوى نظام التشغيل وبذلك يمكن لجميع البرمجيات التي تعمل ضمن هذا النظام الإفادة من هذا الدعم والتعامل مع التواريخ الهجرية بتعديلاتٍ بسيطةٍ جداً.

يعتبر لينكس أكثر أنظمة التشغيل مفتوحة المصدر انتشاراً، وقد تمت بالفعل إضافة دعم التقويم الهجري إلى لينكس بفضل جهود بعض المطورين العرب.

من المتطلبات الأخرى التي لا بدّ من تلبيتها للحصول على تعريبٍ كاملٍ للبرمجيات مفتوحة المصدر هي تطوير مدقق إملائي للغة العربيّة، فبعد أن أصبحت البرمجيات مفتوحة المصدر قادرةً على التعامل مع الأحرف والكلمات العربيّة، يجب أن توفّر هذه البرمجيات إمكانيات التدقيق الإملائي للغة العربيّة لتصبح قادرةً على منافسة البرمجيات التجارية التي تحتوي على هذه الإمكانيات.

لسوء الحظ لا يوجد حالياً مدقق إملائي للغة العربيّة مفتوح المصدر، ولكن هناك العديد من المشاريع والأفكار الطموحة للبدء في تطوير هذا المدقق، ولربما تكون هذه المشاريع قد أثمرت في الوقت الذي ستقرأ فيه هذه الكلمات. فبدون إمكانيات التدقيق الإملائي (والتدقيق النحوي في المستقبل) سيبقى تعريب البرمجيات مفتوحة المصدر منقوصاً.

اليونيكود كحلٍ لمشاكل تعريب البرمجيات مفتوحة المصدر

لقد تسببت الإنترنت وثورة تقنية المعلومات والإتصالات بتغييرٍ جذريٍ في ملامح صناعة تقنية المعلومات في جميع أنحاء العالم، ففي البداية كانت الأنظمة الحاسوبية مجرد جزرٍ مستقلةٍ لا تتخاطب مع بعضها البعض إلا فيما ندر، ولم تظهر في ذلك الحين متطلبات توحيد معايير التواصل وخزن البيانات نتيجةً لذلك. ثم جاءت الإنترنت لتجعل التخاطب والتفاعل بين الأنظمة المختلفة الحجر الأساس في عالم الحوسبة، وبرزت الحاجة أكثر فأكثر إلى توحيد معايير الإتصال وأساليب خزن ومعالجة البيانات في الأنظمة الحاسوبية المختلفة.

وقد أدّى غياب التنسيق بين الجهات المختلفة ومحاولات إحتكار التقنية من قبل بعض الشركات التجارية إلى ظهور عدد كبير من أطقم المحارف للغات المختلف، بل تعدّى الأمر ذلك إلى وجود أكثر من طاقم محارفٍ واحدٍ للغة الواحدة، وهو ما تسبب في الكثير من الصعوبات أثناء محاولة ربط الأنظمة المتخلفة ببعضها البعض، وبدأت الحلول الجزئية بالظهور كتطوير أدوات التحويل بين أطقم المحارف المختلفة، ولكن هذه الحلول تعتمد على محاولة إصلاح أثر المشكلة بدلاً من حل المشكلة من أساسها.

يضاف إلى ذلك عدم توافق أطقم المحارف المختلفة مع بعضها البعض، وهو ما يستدعي القيام بعمليات (الترجمة من طاقم محارفً إلى آخر) أثناء نقل البيانات من نظام إلى آخر، وهو ما قد يتسبب في تحريف أو تغيير هذه البيانات نتيجة عدم توافق أطقم المحارف المختلفة.

من هنا ظهرت الحاجة إلى طريقةٍ معياريةٍ موحدة للتعامل مع جميع لغات العالم بطريقةٍ قياسيةٍ تلغي الحاجة إلى الحلول الجزئية وتضمن تواصل وتفاعل البرمجيات والأنظمة المختلفة في عصر الإنترنت بغضّ النظر عن أماكن تواجد هذه الأنظمة جغرافياً.

تلبيةً لهذه الحاجة، تم تطوير معايير اليونيكود Unicode كحلٍ جذريٍ لمشاكل التخاطب والتواصل في عصر المعلومات، وهي أنظمة ترميز معيارية تقوم بترميز كل حرف من أية لغةٍ من لغات العالم بمقابلٍ وحيدٍ في طاقم محارف اليونيكود، بغضّ النظر عن النظام أو البرنامج الذي يستخدم هذا الحرف. وهو ما يعني بالتالي التخلص من مشاكل عدم التوافق بين أطقم المحارف المختلفة نتيجةً لاستخدام طاقم محارفٍ معياريٍ موحد، ويلغي الحاجة إلى (الترجمة) بين أطقم المحارف المختلفة، وضمان عدم تحريف أو تغيير البيانات أثناء انتقالها من تطبيقٍ لآخر أو من نظامٍ إلى نظامٍ آخر.

عدا عن ذلك، فإن تطوير البرمجيات باستخدام معايير اليونيكود سيؤدّي إلى توفير هذه البرمجيات بجميع لغات العالم دون الحاجة إلى إعادة كتابة هذه البرمجيات من جديد لتلبّي متطلبات كلّ لغةٍ على حدة، ولذلك فإن اليونيكود يعتبر من أهم إنجازات صناعة تقنية المعلومات على الإطلاق.

أدركت الشركات المطورة للبرمجيات التجاريّة أهمية اليونيكود وما يقدّمه من تسريعٍ في تطوير البرمجيات وتخفيضٍ في نفقات التطوير، وبدأت باستخدام هذه المعايير في البرمجيات التي تقوم بتطويرها لتوفير دعم اللغات المختلفة ضمن هذه البرمجيات. وتبنى معظم أنظمة التشغيل التجارية حاليا باستخدام معايير اليونيكود.

تقدّم معايير اليونيكود بالإضافة إلى طاقم المحارف المعياري الموحّد أدواتٍ لتوفير دعم ثنائية الإتجاه وتحليل السياق والرموز الخاصّة والأرقام، مما يجعلها منها حلاً مثالياً للمشاكل التي تواجه اللغة العربيّة أثناء تعريب البرمجيات مفتوحة المصدر.

ونتيجةً للتبني العالمي لمعايير اليونيكود، فقد تم بناء بيئات التطوير المستخدمة في بناء البرمجيات مفتوحة المصدر إعتماداً على هذه المعايير (كبيئة Qt و GTK+)، هذا يعني أن أي برنامجٍ سيطور باستخدام أيٍ من هذه البيئات سيحتوي تلقائياً على دعم معايير اليونيكود وبالتالي دعم اللغة العربيّة (إضافةً إلى غيرها من اللغات).

إن استخدام معايير اليونيكود لتعريب البرمجيات مفتوحة المصدر سيوّفر الكثير من الوقت والعمل لإعادة بناء هذه البرمجيات، وسيتيح توفير النسخ المعرّبة من هذه البرمجيات في نفس وقت إصدار النسخ الأساسية. وسيضمن أيضاً بناء قواعد المعطيات العربيّة بصيغةٍ معياريةٍ موحدّةٍ تسهّل التعامل معها وتخاطبها فيما بينها ومع غيرها في المستقبل. كما سيضمن إمكانيات نقل البيانات عبر البرمجيات والأنظمة المختلفة دون خشية تحريفها أو تلفها مهما تعددت الشركات الصانعة للأنظمة والبرمجيات والدول التي تمر من خلالها هذه البيانات.

أين نحن الآن؟

يعتمد قرار التحوّل إلى البرمجيات مفتوحة المصدر لأي مؤسسةً أو شركةٍ في الوطن العربي على مستوى دعم اللغة العربية المتوفر في هذه البرمجيات، ومدى توفّر البدائل للبرمجيات المغلقة في عالم المصادر المفتوحة. ولذلك لا بدّ من استعراض الوضع الراهن

لدعم اللغة العربيّة في البرمجيات مفتوحة المصدر ليتمكّن صنّاع القرار من اتخاذ القرارات الملائمة لظروف مؤسساتهم وشركاتهم.

ترافقت ثورة برمجيات المصادر المفتوحة منذ البداية بالكثير من محاولات تعريب هذه البرمجيات، والتي استخدمت عدّة أساليب لإضافة دعم اللغة العربيّة إلى هذه البرمجيات، فمنها من اعتمد على إضافة دعم اللغة العربيّة إلى أنظمة التشغيل لتكون متاحةً بشكل شفاف لجميع البرمجيات التي تعمل ضمن هذا النظام، ومنها من عمل على إضافة دعم اللغة العربيّة إلى البرمجيات نفسها لتدعم اللغة العربية بغضّ النظر عن نظام التشغيل الذي تعمل ضمنه.

ونتيجةً لطبيعة منهجية التطوير الخاصّة بالمصادر المفتوحة، فقد تكرر العمل في بعض المشاريع نتيجة غياب التنسيق بين جهات التطوير المختلفة، كما ظهرت عدّة آراءٍ متضاربةٍ نتج عنها عدّة تطبيقاتٍ للتعريب غير متوافقةٍ فيما بينها. إلى أن ظهر اليونيكود والذي غيّر بشكل جذري أسلوب التعامل مع اللغة العربيّة في البرمجيات مفتوحة المصدر.

كما ذكرنا سابقاً فإن معايير اليونيكود وبناء البرمجيات مفتوحة المصدر باعتمادها يعني ضمان الحصول على النسخ المعرّبة مباشرة بعد الإنتهاء من تطوير هذه البرمجيات، ولكن يبقى هناك القليل الذي ينبغي عمله لهذه البرمجيات لتصبح معرّبةً بالكامل، وهو تعريب واجهة المستخدم.

حيث أنّ البرمجيات المبنية وفقاً لمعايير اليونيكود قادرة على التعامل مع النصوص العربيّة، وبإمكانها معالجة الأحرف والكلمات العربيّة، ولكن واجهات هذه البرمجيات ليست عربيّة،

وهو ما قد يشكل عائقاً لدى استخدامها من قبل مستخدمين ليسوا على درايةٍ كافيةٍ باللغة الأجنبية المستخدمة في واجهة هذه البرمجيات.

إن تعريب واجهة المستخدم سهل للغاية من الناحية التقنية، ولكنّه يتطلب جهوداً من المطورين والمترجمين العرب لترجمة نصوص الأوامر والتعليمات في واجهة المستخدم، إضافةً إلى ملفات المساعدة وأدلّة الإستخدام.

لقد قامت مجموعات من المتطوعين بترجمة واجهات الإستخدام للعديد من البرمجيات مفتوحة المصدر، بعضها أصبحت واجهته مترجمةً بالكامل، وبعضها الآخر ما زال في المراحل الجزئية.

إن الحجم الهائل للبرمجيات مفتوحة المصدر، وطبيعة تطوير هذه البرمجيات ذات الوتيرة السريعة يتطلب البدء مباشرةً في العمل على ترجمة وتعريب أكبر قدرٍ ممكنٍ من هذه البرمجيات للإفادة منها عند الإنتقال إلى المصادر المفتوحة.

عند اتخاذ قرار الإنتقال إلى المصادر المفتوحة ينبغي تحديد إحتياجات كلّ مؤسسةٍ على حدة، ومن ثم تقييم البرمجيات المفتوحة المصدر المقترحة من حيث تلبيتها لهذه المتطلبات، ففي حين قد تتطلب بعض المؤسسات توفّر واجهة استخدام عربيّة لجميع البرمجيات التي ستستخدمها، قد تكتفي أخرى بدعم اللغة العربيّة ضمن هذه البرمجيات دون الحاجة إلى واجهات استخدامٍ عربيّة.

أسئلة عليك إجابتها:

عندما ستواجه بقرارٍ عليك اتخاذه فيما يتعلّق بتحوّل مؤسستك إلى المصادر المفتوحة عليك أن تحدّد أيضاً المتطلبات اللغوية لمؤسستك وقدرة البرمجيات مفتوحة المصدر على تلبية هذه المتطلبات. عليك أن تطرح على نفسك الأسئلة التالية:

- ما هو مستوى دعم اللغة العربيّة الذي تتطلبه طبيعة العمل في المؤسسة؟
- هل يكفي أن توفّر البرمجيات مفتوحة المصدر دعم اللغة العربيّة، أم أن وجود واجهة استخدام عربيّة لهذه البرمجيات يعتبر أساسياً؟
- هل تتطلب طبيعة العمل في المؤسسة وجود إمكانيات التدقيق الإملائي في البرمجيات المستخدمة؟
- هل سأحتاج إلى أدلّة استخدامٍ أو ملفات مساعدةٍ باللغة العربيّة للبرمجيات التي أقوم باعتمادها؟ هل هذه الأدلّة أو الملفات متوفرة؟
- هل أملك القدرة على تعريب بعض الواجهات أو أدلّة الإستخدام التي أحتاجها ضمن المؤسسة؟

الفصل الرابع
التفاعل بين البرمجيات المفتوحة والمغلقة

مقدّمة

أثناء محاولتك لاتخاذ قرارٍ يتعلق باعتماد البرمجيات مفتوحة المصدر أو البرمجيات المغلقة، وربما أثناء قراءتك لصفحات هذا الكتاب، قد تقع في حيرةٍ من أمرك حيال هذا الموضوع، فالإقدام على قرار التحوّل بالكامل نحو المصادر المفتوحة مرةً واحدةً قد لا يكون القرار الملائم لوضع مؤسستك الراهن، وفي الوقت ذاته فإنّك لا تريد إغفال ثورة البرمجيات مفتوحة المصدر والبقاء بالتالي في قفص البرمجيات المغلقة وشروط الشركات الإحتكارية.

كما أنّه من الممكن أن لا تجد في عالم البرمجيات مفتوحة المصدر كل حاجتك من التطبيقات البرمجية التي تستخدمها لأداء أعمالك، أو قد تجد نفسك في مواجهة كمٍ هائلٍ من البيانات المحفوظة بصيغ غير معيارية مما قد يضطرك لاستخدام برنامجٍ مغلقٍ أو أكثر للتعامل مع هذه البيانات.

إذا كان هذا الكلام يصف حالك، فلا تهتم، بإمكانك استخدام البرمجيات مفتوحة المصدر جنباً إلى جنبٍ مع البرمجيات المغلقة وبسهولة مطلقة، لقد تم تطوير الكثير من الأدوات والبرمجيات التي ستسهل عليك العمل ضمن بيئةٍ مختلطةٍ من البرمجيات مفتوحة المصدر والبرمجيات المغلقة.

سنحاول في هذا الفصل إلقاء الضوء على أساليب تشغيل البرمجيات مفتوحة المصدر إلى جانب البرمجيات المغلقة، وكيفية تمكينها من التفاعل مع بعضها البعض.

تثبيت عدّة أنظمة تشغيلٍ على نفس الحاسوب

قد تحتاج قبل إتخاذ قرارٍ بالتحوّل إلى البرمجيات مفتوحة المصدر إلى التعرّف بشكل أفضل على خيارات المتاحة من البرمجيات مفتوحة المصدر لتقييم قدرة هذه البرمجيات على تلبية متطلباتك، وقد تحتاج إلى مرحلةٍ إنتقاليةٍ يتمكن خلالها المستخدمون في الشركة من استخدام البرمجيات مفتوحة المصدر أو المغلقة على حدّ سواء، وقد تتخذ قراراً نهائياً يستوجب تواجد البرمجيات المفتوحة المصدر والمغلقة معاً في الحلّ النهائي.

في كلّ هذه الحالات ستحتاج إلى تثبيت عدّة أنظمة تشغيلٍ على نفس الحاسوب، فعلى الرغم من وجود بعض البرمجيات مفتوحة المصدر التي تستطيع العمل ضمن أنظمة التشغيل المغلقة فإن معظم البرمجيات المغلقة لا تعمل ضمن أنظمة التشغيل مفتوحة المصدر (إلا في حال إستخدام برمجياتٍ أو أدواتٍ خاصّة)، عذا عن وجود بعض البرمجيات مفتوحة المصدر التي لا تعمل ضمن أنظمة التشغيل المغلقة.

هناك عدّة أساليب لتثبيت أكثر من نظام تشغيل على نفس الحاسوب، وسنلقي نظرةً فيما يلي على الأساليب المتاحة لتثبيت أكثر أنظمة التشغيل المغلقة ومفتوحة المصدر شعبيةً (ويندوز ولينكس) جنباً إلى جنب. وتطبّق المفاهيم التي سنتعرض لها على أنظمة التشغيل الأخرى مع بعض التعديلات البسيطة.

إن أول مراحل تثبيت لينكس جنباً إلى جنب مع نظام تشغيل آخر تبدأ بتقسيم القرص الصلب لإتاحة مساحةٍ ضمنه لتثبيت لينكس، وهي المرحلة التي تسبب غالبية المشاكل لمستخدمي لينكس الجدد، حيث أن الخطأ هنا قد يتسبب بضياع جميع المعلومات المخزّنة على القرص الصلب، ولذلك لا بدّ من توخّي الحذر عند هذه المرحلة والتأكد من وجود نسخة إحتياطية لجميع الملفات الهامّة. تقدّم غالبية توزيعات لينكس الحديثة أدواتٍ خاصّةً للمساعدة في تجاوز هذه المرحلة بأمان. بعد القيام بتحضير القرص الصلب لإستضافة لينكس سيقوم برنامج الإعداد بتثبيت نظام التشغيل إضافةً (وحسب الخيارات التي تقوم بها أثناء الإعداد) إلى مجموعة من أدوات التطوير والألعاب والبرمجيات.

وفي المرة التالية التي ستقوم بها بتشغيل حاسبك سيقوم بسؤالك عن نظام التشغيل الذي ترغب بإقلاعه، بهذه الطريقة يمكنك العمل ضمن لينكس أو ويندوز متى شئت، ولن يؤثر أي من أنظمة التشغيل على الآخر، فلكل نظامٍ جزء خاص به من القرص الصلب يستخدمه لحفظ إعداداته والتطبيقات التي تعمل ضمنه ومعلومات المستخدم.

الوصول إلى الملفات الخاصّة بكل نظام تشغيلٍ من النظام الآخر

بعد أن تقوم بتثبيت أنظمة التشغيل التي تشاء قد تحتاج إلى الوصول إلى بعض الملفات المخزّنة ضمن أحد أنظمة التشغيل أثناء عمل ضمن النظام الآخر، في الأحوال العادية عليك أن تقوم بإعادة تشغيل جهازك واختيار بدء تشغيل النظام الآخر لتتمكن من الوصول إلى هذه الملفات، ولكنك أحياناً قد تحتاج لمعالجة هذه الملفات ضمن نظام التشغيل الآخر وهنا لن تفيد عملية إعادة الإقلاع وبدء تشغيل النظام الثاني.

يقوم كل نظام تشغيلٍ بتهيئة الجزء المخصص له من القرص الصلب بصيغةٍ خاصةٍ بهذا النظام، ويستخدم نظام ملفاتٍ خاصاً به أيضاً ضمن هذا الجزء (نظام الملفات هو الطريقة التي يقوم نظام التشغيل بتنظيم توضّع الملفات ضمن القرص الصلب).

لقد تم تطوير بعض الأدوات التي تتيح لك الوصول إلى أنظمة الملفات الخاصّة بأنظمة التشغيل الأخرى من داخل ويندوز أو لينكس، فإذا كنت تعمل ضمن ويندوز وأردت استخدام ملفٍ مخزّنٍ ضمن نظام ملفات لينكس يمكنك إستخدام أدواتٍ مثل FSDEXT2 والتي تتيح لك قراءة الملفات فقط أو Explore2fs التي تتيح لك قراءة وتعديل الملفات، هذه الأدوات مجانيّة وبإمكانك الحصول عليها عبر الإنترنت دون مقابل.

أما إذا كنت تعمل ضمن لينكس وأردت الوصول إلى ملفاتٍ مخزّنةٍ ضمن نظام ملفات ويندوز فإن العملية أسهل بكثير، يحتوي لينكس بشكل إفتراضي على دعمٍ لأنظمة ملفات ويندوز، ولأن أنظمة ويندوز المختلفة تستخدم عدّة أنظمة ملفات فإن أسلوب الوصول إلى هذه الملفات يعتمد على نظام الملفات المستخدم، والتي قد تتطلب القيام ببعض الإعدادات البسيطة ضمن لينكس.

ويدعم لينكس إضافةً إلى أنظمة ملفات ويندوز عدداً كبيراً من أنظمة الملفات، أي أنّه يستطيع التكلم بلغة عددٍ كبيرٍ من أنظمة التشغيل والوصول إلى أنظمة الملفات الخاصّة بها، وهي من الميزات الهامّة لنظام التشغيل لينكس والتي تجعل تفاعله مع أنظمة التشغيل الأخرى أكثر سهولةً.

برمجيات المواءمة

قد تضطر إلى الحفاظ على برنامجٍ مغلقٍ معينٍ بسبب عدم توفّر بديل مفتوح المصدر له أو بسبب وجود حجمٍ هائلٍ من البيانات المخزّنة بالصيغة الخاصّة بهذا البرنامج والتي لا يمكن تحويلها بسهولة. وقد تجد أنه من غير المنطقي أن تحتفظ بنظام التشغيل المغلق لجرد حاجتك إلى تشغيل برنامج أو برنامجين.

تستخدم برمجيات المواءمة لتشغيل التطبيقات البرمجية المصممة للعمل ضمن نظام تشغيلٍ معين مع نظام تشغيل آخر. سنستعرض فيما يلي برمجيات المواءمة المتوفّرة لتشغيل البرمجيات المصممة للعمل ضمن ويندوز في لينكس. لا توجد برمجيات مواءمةٍ لتشغيل تطبيقات لينكس ضمن ويندوز، ولكنّ معظم البرمجيات مفتوحة المصدر والتي تعمل ضمن لينكس تستطيع العمل ضمن ويندوز أيضاً.

يعتبر مشروع WINE من أهم البرمجيات التي تساعد على تشغيل برمجيات ويندوز ضمن لينكس، وهو برنامج مفتوح المصدر ويمكن الحصول عليه مجّاناً عبر الإنترنت، و WINE ليس برنامج مواءمةٍ بالمعنى الحرفي للكلمة، فهو يوفّر دعماً منخفض المستوى لبرمجيات ويندوز، مما يجعله أكثر سرعةً في الأداء في مقابل برمجيات المواءمة.

لا بدّ من التحقق من دعم WINE للبرمجيات التي سيتم استخدامها ضمنه، فبعض البرمجيات قد لا تكون متوافقةً بالكامل مع WINE. يستخدم WINE من قبل شركة Corel لتمكين برمجياتها من العمل ضمن لينكس.

هناك أيضاً برنامج VMWare والذي يتيح تشغيل نظام تشغيلٍ ضمن نظام تشغيلٍ آخر، وهو يعمل ضمن لينكس وويندوز، هذا البرنامج ليس برنامج مواءمة فهو يقوم بتشغيل نسخة فعليّة من نظام التشغيل (الضيف) ضمن نظام التشغيل (المضيف). VMWare ليس مجانياً وينبغي عليك أن تدفع رسوم ترخيصٍ للشركة المنتجة لقاء استخدامه.

يتيح VMWare للبرمجيات التي تعمل ضمن كلٍ من نظامي التشغيل الضيف والمضيف مشاركة البيانات فيما بينها، كما يمكّن المستخدم من القيام بعمليات النسخ واللصق فيما بين نظامي التشغيل.

هناك مشروع لتطوير برنامج مفتوح المصدر يقابل VMWare ويدعى FreeMWare ويهدف إلى إتاحة تشغيل ويندوز من داخل لينكس (www.freemware.org).

ولا بدّ أيضاً من ذكر برنامج Win4Lin والذي يتيح لمستخدمي لينكس تشغيل ويندوز ضمن لينكس وليس العكس، وهو أسرع من VMWare ويتطلّب حجم ذاكرةٍ أقل، وهو أيضاٌ برنامج تجاري غير مجاني.

لا بدّ من التنويه إلى أن استخدام برمجيات المواءمة لتشغيل برمجيات ويندوز ضمن لينكس لا يلغي الحاجة إلى الحصول على ترخيص إستخدامٍ لهذه البرمجيات، تأكّد من حصولك على الترخيص اللازم لتشغيل أية برمجياتٍ تجاريةٍ حتّى لو كنت تريد تشغيلها ضمن لينكس (بما في ذلك نظام التشغيل ويندوز).

تطبيقات الوب

يعتبر دعم لينكس الأساسي لبروتوكول الإنترنت TCP/IP إضافةً إلى وثوقيته وتكلفته المنخفضة سبباً رئيسياً لاختياره كمخدّم، وتشير الإحصاءات إلى أن لينكس هو أكثر أنظمة تشغيل المخدّمات شعبيةً على الإنترنت. هناك أيضاً أباتشي Apache الذي يعتبر أكثر مخدّمات الوب إنتشاراً.

وفي المقابل يحتل ويندوز مركز الصدارة في مجال البرمجيات المكتبية، ونظراً لإنتشاره الواسع في أوساط المستخدمين فإن معظم شركات تطوير البرمجيات غالباً ما تصدر نسخً من برامجها لتعمل ضمن ويندوز قبل أن تقوم بإصدار نسخٍ لأنظمة التشغيل الأخرى.

لن نبحث في تفاصيل جميع تطبيقات الوب لأن هذا الموضوع وحده قد يحتاج عشرات الكتب، بل سنحاول تغطية بعض النقاط المتعلقة بتفاعل البرمجيات مفتوحة المصدر والبرمجيات المغلقة على الإنترنت.

توجد حالياً ثلاثة مخدّمات شهيرة للوب: المخدّم مفتوح المصدر أباتشي Apache ومخدّم Internet Information Server من مايكروسوفت ومخّدم الوب من Netscape.

يعمل أباتشي ضمن نظام التشغيل لينكس، كما تتوفر منه نسخ تعمل ضمن ويندوز، أما IIS فيأتي مضمنّاً في بعض إصدارات ويندوز وهو يعمل فقط ضمن ويندوز، في حين يتوفر Netscape في نسخٍ تعمل ضمن ويندوز ولينكس.

بما أنّ معايير الإنترنت القياسية موضوعة من قبل جهة مرجعية واحدة (IETF) فإن جميع مخدّمات الوب تعمل بطريقة قياسية ولا يمكن التمييز بينها من وجهة نظر المستخدم على الإنترنت، وتظهر المشكلتين الأساسيتين في التوافقية عند الحاجة إلى الربط مع قواعد البيانات وبناء صفحات الوب الديناميكية.

تقوم معظم المواقع الكبيرة بتخزين محتوياتها في قواعد للبيانات وتستخدم هذه البيانات لبناء صفحات الموقع بشكل ديناميكي. توجد عدّة أدواتٍ برمجية لبناء صفحات الوب بشكل ديناميكي، من أشهر هذه الأدوات PHP و Active Server Pages (ASP) .

PHP هو برنامج مفتوح المصدر مصمم للعمل مع لينكس ومخدّم الوب أباتشي، وتتوفّر أيضاً نسخة منه للعمل ضمن ويندوز (www.php.net).

هناك بعض الإعتبارات لربط PHP مع قواعد البيانات المغلقة، في بيئة ويندوز يمكن إستخدام أدوات ODBC للربط مع قاعدة البيانات، أما الربط مع قواعد البيانات المغلقة ضمن لينكس فهو أكثر تعقيداً، ولكنّه ممكن.

يتضمن مخدّم الوب IIS من مايكروسوفت برنامج ASP لبناء صفحات الوب ديناميكياً، يمكن ربط ASP مع قواعد البيانات مفتوحة المصدر باستخدام أدوات ربط قواعد البيانات ODBC.

قد يبنى موقع شركةٍ ما على الإنترنت باستخدام ASP على سبيل المثال، هذه الشركة ترغب بالتحوّل إلى نظام التّشغيل لينكس، بإمكان هذه الشركة الحفاظ على الموقع الذي قامت

بتطويره وتشغيلها ضمن لينكس باستخدام برمجيات مثل Chili!Soft. وقد ترغب الشركة أيضاً في تحويل موقعها من ASP إلى PHP للإفادة من مجانيّة PHP وتوفرها على كلٍ من ويندوز ولينكس، هناك برنامج مجاني يدعى ASP2PHP يمكن استخدامه لإنجاز هذا التحويل بسرعةٍ وسهولة.

يتوفر متصفح الوب Netscape في نسخٍ تعمل ضمن كلٍ من لينكس وويندوز، وهو يدعم جميع المعايير القياسية، ويوزّع مع معظم توزيعات لينكس، كما يمكن الحصول على نسخةٍ تعمل ضمن ويندوز من الموقع التالي: www.netscape.com. وقد قامت Netscape بتوزيع الشيفرة المصدرية لمتصفحها لتطوير متصفحٍ جديدٍ يدعى Mozilla، نتيجة لطبيعة تطوير برمجيات المصادر المفتوحة فقد أصبح Mozilla أسرع وأكثر وثوقيةً من Netscape ، يمكن الحصول على نسخةٍ من هذا المتصفح تعمل ضمن لينكس أو ويندوز من الموقع التالي www.mozilla.org.

هناك أيضاً بعض المتصفحات التي تعمل ضمن ويندوز فقط، أكثر هذه المتصفحات المغلقة شعبيةً متصفح Internet Explorer الخاص بمايكروسوفت، وهو يدعم جميع المعايير القياسية إضافةً إلى بعض الميزات الخاصّة بمايكروسوفت (ActiveX و VBScript).

التطبيقات المكتبيّة

ونقصد هنا بالتطبيقات المكتبيةّ البرمجيات التي يتم تثبيتها ضمن حاسوب المستخدم لتوفير خدماتٍ تساعده في أداء مهامه في العمل. وتحتوي حزم البرمجيات المكتبية الشهيرة مجموعةً من التطبيقات المكتبيّة، كحزمة مايكروسوفت أوفيس التي تحتوي على برامج لتحرير النصوص والجداول الحسابية وقواعد البيانات والعروض التقديمية.

تعتبر حزمة برمجيات مايكروسوفت أوفيس أكثر حزم التطبيقات البرمجية شعبيةً على الإطلاق، ونتيجةً لهذه الشعبية فإن معظم الملفات التي ستصلك من جهات أخرى ستأتيك بصيغة مايكروسوفت أوفيس. ولذلك ولكي يحتلّ لينكس مكانته الطبيعية في عالم التطبيقات المكتبية لا بدّ من توفّر حزمة برمجياتٍ مشابهة لمايكروسوفت أوفيس بإمكانها التعامل مع صيغ ملفات أوفيس.

تشكّل حزمة التطبيقات المكتبية ستار أوفيس StarOffice والتي تنتجها شركة Sun البديل الأمثل لمايكروسوفت أوفيس ضمن لينكس، تحتوي حزمة ستار أوفيس برامج لتحرير النصوص والجداول الحسابية والعروض التقديمية وقواعد البيانات وتحرير الرسوميات، وتدعم صيغ ملفات أوفيس بشكل جيد، أي أنك تستطيع قراءة وحفظ ملفات مايكروسوفت أوفيس ضمن ستار أوفيس، وهو ما يقدّم حلاً لمشكلة توافقية الملفات الموجودة حالياً في المؤسسة أو التي تأتي من خارج المؤسسة.

لقد توقفت شركة Sun عن توزيع ستار أوفيس كبرنامجٍ مفتوح المصدر لإعتباراتٍ تتعلق بترخيص بعض أجزاء شيفرته المصدرية، ولكن هناك برنامجاً مفتوح المصدر مقابل لستار أوفيس، ويحتوي نفس إمكانيات ستار أوفيس تقريباً، ويمكن الحصول عليه مجّاناً عبر الإنترنت (www.openoffice.org). وقد قطع دعم اللغة العربيّة ضمنه مرحلةً جيدةً، ولكنه مازال يفتقر إلى إمكانيات التدقيق الإملائي. تتوفر نسخ من كلٍ من هذين البرنامجين للعمل ضمن لينكس أو ويندوز.

تتوفّر أيضاً حزم تطبيقاتٍ مكتبيةٍ أخرى مفتوحة المصدر تعمل ضمن نظام التشغيل لينكس، وهي تدعم صيغ ملفات مايكروسوفت أوفيس بدرجاتٍ متفاوتة، من أهم هذه البرمجيات حزمتي Koffice و Gnome Office.

قواعد البيانات

تتزايد أهمية قواعد البيانات في عالم الأعمال مع إزدياد إعتماد هذه الأعمال على الحواسيب والإنترنت لأداء مهامها والحفاظ على قدراتها التنافسية، ونتيجةً لهذه الأهمية المتزايدة فقد تمّ تطوير العشرات من برمجيات إدارة قواعد البيانات المغلقة والمفتوحة المصدر. تقدّم الشركات المنتجة لبرمجيات إدارة قواعد البيانات الشهيرة نسخاً من برمجياتها للعمل ضمن كلٍ من لينكس وويندوز، من أمثلة هذه البرمجيات المغلقة Oracle و Adabas و Sybase، ومن أشهر برمجيات إدارة قواعد البيانات مفتوحة المصدر MySQL و PostrgreSQL.

وتنتج مايكروسوفت برنامجين لإدارة قواعد البيانات ولكنهما لا يعملان إلى ضمن نظام التشغيل ويندوز: Access و SQL Server.

تستخدم برمجيات إدارة قواعد البيانات من مايكروسوفت أدوات الربط مع قواعد البيانات Open Data Base Connectivity، ولقد تم تطوير عدّة برمجيات مفتوحة المصدر لتوفير إمكانيات ربط قواعد البيانات مفتوحة المصدر مع تلك المغلقة، من هذه البرمجيات MyODBC الذي يوفّر إمكانيات ربط برنامج إدارة قواعد البيانات MySQL مع قواعد بيانات Access أو SQL Server.

تشغيل البرمجيات عبر الشبكة

ظهر منذ عهدٍ قريب مفهوم الحاسوب الشبكي Network Computer والذي يعتمد على مبدأ تشغيل جميع التطبيقات البرمجيّة التي سيحتاجها المستخدم على المخدّم وتحويل حاسوب المستخدم إلى مجرّد أداةٍ لعرض وإدخال البيانات، لا يتطلب هذا الأسلوب وجود قدراتٍ حاسوبيةٍ كبيرة على جهاز المستخدم، لذلك فإن هذا الحل يوفّر الكثير من تكاليف أنظمة تقنية المعلومات، كما يسهّل مهام تشغيل وإدارة هذه الأنظمة نتيجة مركزية جميع التطبيقات والإعدادات.

قامت شركة Citrix بتطوير برنامجٍ لتوفير هذه الإمكانيات يدعى MetaFrame، وهو يتيح تشغيل البرمجيات ضمن ويندوز وتفاعل المستخدم مع هذه البرمجيات من داخل لينكس، وقد أدخلت مايكروسوفت هذه التقنية ضمن أنظمة التشغيل التي تنتجها ولكنّها أكثر محدوديةً ولا توفّر إمكانية التفاعل مع البرمجيات من داخل لينكس.

إن استخدام تقنية Citrix التي تسمى Independent Computing Architecture (ICA) يتيح إعداد البرمجيات المطلوبة مرةً واحدةً على المخدّم ومن ثمّ استخدامها من قبل أي عددٍ من المستخدمين سواء كانوا يستخدمون ويندوز أو لينكس على حواسبهم. لا بدّ من التنويه إلى ضرورة الحصول على العدد الملائم من تراخيص الإستخدام لأي برنامجً تريد تثبيته على المخدّم والتفاعل معه عبر الشبكة باستخدام MetaFrame، فعلى الرغم من أنك ستقوم بتثبيت هذا البرنامج مرةً واحدةً على المخدّم، إلا أنّ عدد المستخدمين الفعلي يتضمن كلّ من سيستخدم هذا البرنامج بواسطة MetaFrame. لا بدّ أيضاً من أخذ تكاليف ترخيص برمجيات MetaFrame بالحسبان، فهي ليست مفتوحة المصدر أو مجانيّة، وغالباً ما يتم

تسعير ترخيص الإستخدام تبعاً لعدد المستخدمين ضمن الشبكة. للحصول على المزيد من المعلومات يمكنك زيارة الموقع التالي على الإنترنت www.citrix.com.

هناك برنامج مفتوح المصدر (Virtual Network Computing VNC) يشبه إلى حدٍ ما البرامج المذكورة من حيث قدرته على تشغيل برنامجٍ ما ضمن حاسوبٍ والتفاعل مع هذا البرنامج من حاسوبٍ أخر (حتى إذا اختلف نظام التشغيل المستخدم على كلٍ منهما)، أي أنّك تستطيع التفاعل مع برنامجٍ يعمل ضمن ويندوز من داخل لينكس والعكس. إلا أن هذا البرنامج لا يملك القدرة على التفاعل مع أكثر من برنامجٍ واحدٍ في الوقت نفسه. هذا البرنامج مجاني ويمكن الحصول عليه من الموقع التالي www.uk.research.att.com/vnc.

هذا البرنامج يشكل حلاً رائعاً للحالات التي قد تجد نفسك فيها مضطراً للإحتفاظ ببرنامجٍ مغلقٍ واحدٍ أو أكثر لاستخدامه من حينٍ لآخر، وعندها تستطيع الإنتقال إلى البرمجيات مفتوحة المصدر مع الحفاظ على هذه البرمجيات المغلقة على حاسوبٍ واحدٍ لتستخدمها عند الحاجة بواسطة VNC.

تشبيك برمجيات المصادر المفتوحة والمغلقة

تختلف أساليب التشبيك بين برمجيات المصادر المفتوحة والمغلقة بشكلٍ كبيرٍ نظراً لاختلافها في النشأة والتطور. تستخدم البرمجيات المغلقة بروتوكولات خاصّة قامت الشركات المنتجة لهذه البرمجيات بتطويرها لتشبيك هذه البرمجيات مع بعضها البعض، وهي في أغلب الأحيان مغلقة أيضاً. أما البرمجيات مفتوحة المصدر فتعتمد للتشبيك فيما بينها على

بروتوكولات قياسية (أهمها بروتوكول الإنترنت TCP/IP). وتستخدم الإنترنت (وهي أكبر الشبكات الحاسوبية على الإطلاق) هذا البروتوكول لنقل البيانات ضمنها (وهو ما يتضح جلياً من إسم هذا البروتوكول).

لقد تمّ تطوير مجموعة من البرمجيات التي تقوم بمواءمة أداء البروتوكولات المستخدمة في تشبيك البرمجيات المغلقة لتمكينها من التواصل مع برمجيات المصادر المفتوحة، كما قامت الشركات المنتجة للبرمجيات المغلقة بإضافة دعم بروتوكول الإنترنت إلى برمجياتها.

سنقوم فيما يلي باستعراض أهم البرمجيات التي توفر دعم التشبيك بين البرمجيات مفتوحة المصدر والمغلقة.

يهدف مشروع سامبا Samba (www.samba.org) إلى بناء مجموعة من البرمجيات التي تتيح للنظام مفتوح المصدر التحدّث بلغة النظام المغلق (ويندوز) وبالتالي تمكين هذين النظامين من التواصل، أي أنّ أيّ نظام تشغيلٍ مفتوح المصدر يستخدم سامبا سيظهر ضمن الشبكة وكأنّه نظام ويندوز، ويستطيع مستخدمو ويندوز الوصول إلى هذا النظام المفتوح عبر الشبكة ومشاركة الملفات والطابعات والعكس صحيح.

يمكن أن يقوم سامبا بمهام أنظمة تشغيل المخدّمات من مايكروسوفت (ويندوز NT وويندوز 2000)، مما يعني أن بإمكانك توفير خدمات مشاركة الملفات والطباعة التي تقدّمها عادةً أنظمة تشغيل ويندوز باستخدام لينكس، أي أنّ المستخدم العادي لن يلحظ اختفاء ويندوز من على المخدّم، في حين ستلحظ الإدارة المالية الوفر في نفقات تراخيص الإستخدام الموافقة

(عدا عن المتطلبات المتواضعة لسامبا من حيث التجهيزات مقارنةً بأنظمة ويندوز التي تقدّم نفس الخدمات).

تحتوي معظم توزيعات نظام التشغيل مفتوح المصدر لينكس على سامبا، مما يجعل عملية تشبيكه مع نظام التشغيل ويندوز أمراً غايةً في السهولة وهو برنامج مفتوح المصدر وبمكن الحصول عليه مجاناً. كما توجد منه نسخ أخرى للعديد من أنظمة التشغيل مفتوحة المصدر بل وحتّى المغلقة (مثل MacOS).

تستخدم أنظمة التشغيل مفتوحة المصدر بروتوكول نظام الملفات الشبكي Network File System (NFS) بشكلٍ أساسي لمشاركة الملفات عبر الشبكة، وهو متوفر أيضاً ضمن معظم أنظمة التشغيل المغلقة. وتحتوي أنظمة التشغيل مفتوحة المصدر على أدواتٍ لإدارة هذا البروتوكول بغية إتاحة مشاركة الملفات بين الحواسيب المختلفة ضمن الشبكة.

لا يحتوي نظام التشغيل ويندوز على دعمٍ لنظام الملفات الشبكي NFS، ولكن توجد بعض البرمجيات التي يمكن تثبيتها ضمن ويندوز لتمكينه من التواصل مع الأنظمة مفتوحة المصدر ضمن الشبكة. معظم هذه البرمجيات مغلقة ويجب دفع رسوم ترخيص الإستخدام للحصول على الحقّ باستخدامها، من هذه البرمجيات على سبيل المثال لا الحصر برنامج NFS Maestor الذي تنتجه شركة Hummingbird وبرنامج Chameleon NFS من شركة NetManage. كما تقوم مايكروسوفت بتطوير حزمة برمجيات Windows for Unix والتي توفّر دعم نظام الملفات الشبكي NFS ضمن أنظمة ويندوز.

هناك أيضاً بعض خدمات التشبيك القياسية مثل بروتوكول نقل الملفات File Transfer Protocol (FTP) والذي يوفّر إمكانية نقل الملفات من جهازٍ لآخر ضمن الشبكة (وكونها قياسية فإنها ستعمل بين أنظمة التشغيل المختلفة). تحتوي معظم أنظمة التشغيل على أدوات للإفادة من هذه الخدمة، ولكنًها لا تحتوي على برنامج المخدّم الذي يوفر الخدمة، ويمكن الحصول على هذا البرنامج مجاناً كبرنامجٍ مفتوح المصدر أو الحصول على نسخة تجارية منه لقاء مقابلٍ مادّي.

أسئلة عليك إجابتها

يعتبر تواجد البرمجيات مفتوحة المصدر والبرمجيات المغلقة ضمن نفس الشبكة وإمكانية تفاعلها مع بعضها أمراً هامّاً للغاية مهما كان مستوى الإعتمادية على المصادر المفتوحة. عليك الإجابة على الأسئلة التالية لتحديد متطلبات تشبيك البرمجيات مفتوحة المصدر والمغلقة ضمن مؤسستك:

- هل أنا بحاجة إلى تثبيت عدّة أنظمة تشغيل على نفس الحاسوب؟
- ما هي متطلبات التفاعل بين أنظمة التشغيل المتعددة المثبّتة على نفس الحاسوب؟
- هل أنا بحاجة إلى بعض البرمجيات المغلقة والتي لم أجد لها بديلاً مفتوح المصدر؟
- ما هي أفضل أساليب الإستمرار في استثمار هذه البرمجيات المغلقة؟
- ما هي التطبيقات التي تعتمد على الوب الموجودة في المؤسسة؟ وما هو مدى توافقها مع برمجيات إستعراض الإنترنت المختلفة؟
- هل أنا بحاجة لتشغيل بعض التطبيقات عبر الشبكة؟

الفصل الخامس

البعد الإقتصادي للمصادر المفتوحة

مقدّمة

ذكرنا في الفصول السابقة أن هناك فهماً مغلوطاً للبرمجيات مفتوحة المصدر على أنها مجانيّة، ومع أن بعض البرمجيات مفتوحة المصدر قد تكون مجانيّة، إلا أن التعريف الصحيح لهذه البرمجيات هو أنّها (حرّة) وليست (مجانية)، بمعني أن مستخدم هذه البرمجيات يملك مطلق الحرية في الحصول على الشيفرة المصدرية لهذه البرمجيات وتعديلها وإعادة توزيعها.

هذا الفهم المغلوط تسبّب في الكثير من الفوضى في مختلف الأوساط المرتبطة بالمصادر المفتوحة، وأدّى إلى نشوب حربٍ شعواءٍ بين أنصار البرمجيات مفتوحة المصدر ومطوري البرمجيات التجارية (والتي غالبا ما تكون مغلقةً). كما تسبب في نشوء خلافات في وجهات النظر ضمن المؤسسات فيما يتعلّق بالجدوى الإقتصادية لاستخدام المصادر المفتوحة.

سنحاول في هذا الفصل تسليط الضوء على الجانب الإقتصادي للبرمجيات مفتوحة المصدر، وسنبحث ذلك من وجهة نظر الشركات المطورة للبرمجيات والشركات والمؤسسة التي ترغب بالتحوّل إلى المصادر المفتوحة. سنستعرض بعض فرص الإستثمار والربح في عالم المصادر المفتوحة، إضافةً إلى المزايا التي تجملها برمجيات المصادر المفتوحة من الناحية الإقتصادية للشركات والمؤسسات الراغبة في التحوّل إلى المصادر المفتوحة.

هل أستطيع الإفادة مادّياً من برمجيات المصادر المفتوحة؟

تشكلت نتيجة الفهم المغلوط لتعريف البرمجيات الحرّة وجهات نظرٍ معاديةٍ لهذه البرمجيات في الأوساط التجارية للظن الخاطئ بأن هذه البرمجيات ستقضي على أرباح هذه الشركات. وارتبطت صورة مطوري البرمجيات مفتوحة المصدر بالأبطال الأسطوريين الذين تقودهم المبادئ والقيم. واتهمت معظم كتابات أنصار البرمجيات الحرّة الشركات التجارية بتخليها عن الهدف الأساسي في تلبية متطلبات زبائنها ببرمجيات عالية النوعية على المدى البعيد مقابل الأرباح المادية على المدى القصير.

ولكن الواقع مختلف تماماً، فحتى ريتشارد ستولمان على سبيل المثال (والذي يعتبر أكثر أنصار البرمجيات الحرّة معاداةً للجوانب التجارية) لم يدعو إلى الإبقاء على نشاطات تطوير البرمجيات مجانيةً أو دون أي مقابلٍ مادّي في جميع الحالات.، ولكنه عارض المعاملة التجارية للبرمجيات على أنّها (نتاج فكري) مطالباً بالتعامل مع تطوير البرمجيات تجارياً كخدمة وليس كمنتج.

لقد لفت النجاح المبهر الذي حققه نظام التشغيل مفتوح المصدر لينكس وتبني العديد من الشركات العالمية سياسةً جديدةً تجاه البرمجيات مفتوحة المصدر والاستثمار في هذا المجال أنظار الجميع إلى الأهمية الإقتصادية للبرمجيات مفتوحة المصدر لكلٍ من الشركات والمؤسسات المستخدمة لهذه البرمجيات والشركات المنتجة لها.

إن البرمجيات الحرّة تتطلب أن يقوم مطوروها بتوفير الشيفرة المصدرية لهذه البرمجيات، ومنح المستخدمين الحقّ في تعديل أو تطوير أو إعادة توزيع هذه البرمجيات بحرية. وهناك

الكثير من مجالات الإستثمار والربح في عالم المصادر المفتوحة، مع الإنتباه إلى أن ذلك لا يعني على الإطلاق بيع (رخص) استخدام هذه البرمجيات لقاء مقابلٍ مادّي.

قد يبدو هذا النقاش للوهلة الأولى غريباً، فنحن هنا نحاول مناقشة مبدأٍ جديدٍ للاستثمار والربح في صناعة البرمجيات في مواجهة مبادئ نشأت منذ عشرات السنين، وتطورت وجرّبت إلى درجة أنّها أضحت تعتبر مرجعاً في الجدوى الإقتصادية لصناعة البرمجيات. ولمواجهة ذلك علينا أن ندرك أن صناعة البرمجيات تمر اليوم في ظروفٍ خاصةٍ جداً، مما قد يتطلب التعامل مع هذه الظروف بأساليب خاصّةٍ أيضاً، فالبرمجيات الحرّة أصبحت حقيقة واقعة، وهي تغيّر من معالم صناعة تقنية المعلومات بأكملها، ولا بدّ من إيجاد أساليب جديدةً للتعامل معها بما يضمن استمرار عمل الشركة وتحقيقها للأرباح.

إقتناص الفرصة

إن التحوّل باتجاه تبني سياسة تطوير تعتمد المصادر المفتوحة قرار مصيري بالنسبة لأي شركةٍ تجاريةٍ، ولذلك لا بدّ لك قبل أن تقدم على أية خطوةٍ في هذا الإتجاه أن تجيب على السؤال التالي: "هل أنا حقاً محتاج إلى ذلك؟". إذا كانت شركتك رابحةً ولا توجد لديك أية مشاكل تريد حلّها أو فرصٍ تريد اقتناصها فدع المصادر المفتوحة وشأنها، ولا تأبه لكل ما ستقرأه في هذا الفصل، إن اعتماد سياسة التطوير باعتماد المصادر المفتوحة سيكلفك الكثير من المال والوقت والعمل، ولا أرى أي حاجةٍ لك في ذلك ما لم يعد عليك ببعض الفوائد.

وقد توجد لديك بالفعل بعض المشاكل التي تريد حلها، أو بعض الفرص التي تريد إقتناصها والإفادة منها في المستقبل، إليك بعض المشاكل أو الفرص التي تواجه شركات تطوير البرمجيات في الوقت الراهن:

- التطوير المستمر للبرمجيات: إن استمرار شركتك في العمل يتطلب ابتكار المزيد من البرمجيات الجديدة أو التطويرات للبرمجيات القديمة بغية الحفاظ على أرباح الشركة.
- نوعية المنتج: يجب أن تكون برمجياتك الجديدة ذات نوعيةٍ عاليةٍ لتكون قادرةً على المنافسة.
- الدعم الفنّي: يجب أن توفر أعلى مستويات الدعم الفنّي لبرمجياتك مع الحفاظ على التطوير المستمر لهذه البرمجيات.
- بناء شراكاتٍ فعّالة: يجب أن تقوم ببناء شبكةٍ من الشراكات لدعم وتطوير وتخديم برمجياتك لزيادة القيمة المضافة إلى هذه البرمجيات بالنسبة للمستخدم والتي ستساعدك بدورها على زيادة مبيعاتك.
- توظيف أمهر المطورين: للحفاظ على نوعية البرمجيات التي تقوم بإنتاجها وتطويرها بشكلٍ مستمر عليك أن تقوم بتحفيز موظفيك بأساليب مبتكرة تتجاوز الوسائل التقليدية، إضافة إلى البحث باستمرارٍ عن الكفاءات التي ستعمل على تطوير منتجاتك في المستقبل.

إن اعتماد سياسة التطوير باستخدام المصادر المفتوحة سيسهم في توفير الدعم لكل من النقاط السابقة، فإذا كنت مهتماً ببعض هذه النقاط فإن هذا الفصل سيزودك بالمزيد من المعلومات العملية حول الاستثمار التجاري للبرمجيات مفتوحة المصدر.

الشيفرة المصدرية: الدجاجة التي تبيض ذهباً!

لنعتبر أنك تنظر الآن إلى المصادر المفتوحة كطريقةً محتملةً للحفاظ على قدرة شركتك التنافسية. لا بدّ أنّك قد واجهت إذاً التساؤل التالي:" إنّ الشيفرة المصدرية التي قمنا بتطويرها على مدى سنوات عدّة تعتبر الدجاجة التي تبيض ذهباً بالنسبة للشركة، ما الذي يدفعني إذاً إلى الإستغناء عن هذه الدجاجة؟"

بمعنى آخر فإن هذه الدجاجة هي كنز الشركة الخفي، والذي يجب الحفاظ عليه وحمايته بكل الوسائل. ولذلك فإن معظم شركات تطوير البرمجيات تتيح الحصول على الشيفرة المصدرية الخاصّة ببرمجياتها ضمن ظروفٍ خاصّةٍ جداً، وغالباً ما يترافق ذلك مع إجراءاتٍ قانونيةٍ خاصّةٍ والكثير من المال في المقابل.

ولكن حماية الشيفرة المصدرية لذاتها أمر غير منطقي، لأن قيمة هذه الشيفرة المصدرية تنبع من القدرة على استثمارها للحصول على منتجاتٍ أو خدماتٍ يمكن بيعها لقاء مقابلٍ مادّي، وهو ما يتطلب عادةً وجود شركةٍ ذات سمعةٍ ومكانةٍ في السوق. كذلك فإن نشر الشيفرة المصدرية لا يعني بالضرورة أنّ الشركات الأخرى ستصبح قادرةً على استنساخ نجاحاتك أو إيذاء أعمالك، فالفضل في هذه النجاحات لا يعود إلى الميزات المتوفرة ضمن الشيفرة المصدرية وحسب، بل هناك العديد من العوامل الأخرى كثقة المستخدمين في شركتك، نوعية الخدمات التي تقدّمها إليهم، وفعالية أقسام التسويق والمبيعات لديك. وأنت من يستطيع التحكم بهذه العوامل حتى عند فقدانك السيطرة على الشيفرة المصدرية ذاتها!

ما هي قيمة الشيفرة المصدرية؟

بعد أن تتقبل مبدأ توزيع الشيفرة المصدرية لواحدٍ أو أكثر من منتجاتك البرمجية كجزءٍ من استراتيجيتك في العمل ستواجه السؤالين التاليين:

1. كيف سيقدّم توزيع الشيفرة المصدرية المزيد من القيمة المضافة إلى زبائني؟
2. كيف ستتمكن الشركة من تحويل هذه القيمة المضافة إلى أرباح مادّية؟

الشيفرة المصدرية لبرنامجٍ ما هي النصوص المكتوبة بلغة البرمجة المستخدمة لبناء هذا البرنامج والتي يتمكن الإنسان من قراءتها وتعديلها واستخدامها لبناء نسخةٍ جديدةٍ من البرنامج أو برنامجٍ آخر يستخدم بعض أو جميع هذه النصوص.

في بداية عصر تقنية المعلومات كانت البرمجيات توزّع بصيغة الشيفرة المصدرية، وهو ما ينعكس إيجاباً على زيادة قيمة هذه البرمجيات بالنسبة للمستخدم عبر تمكينه من تجاوز عقباتٍ عدّةٍ، منها على سبيل المثال:

- يستطيع المستخدم حماية استثماراته في منتجٍ برمجيٍ معين في حال أغلقت الشركة المطورة لهذا البرنامج أبوابها أو قررت التوقف عن دعم هذا البرنامج.
- يستطيع المستخدم التعرّف على آلية عمل هذا البرنامج بشكلٍ أفضل في حال كانت تعليمات الإستخدام المرفقة هزيلةً أو غير مكتملة.
- يستطيع المستخدم متابعة وإصلاح الثغرات الأمنية في البرمجيات والتي قد تتسبب في إلحاق الأذى بأعمال المستخدم.
- يستطيع المستخدم إصلاح المشاكل البرمجية بنفسه في حال عجز أو تمنّع الشركة المنتجة عن ذلك.

- يستطيع المستخدم بناء البرنامج من جديد ليدعم أنظمة تشغيلٍ أو تجهيزاتٍ جديدةً لم توفر الشركة المنتجة دعماً لها.
- يستطيع المستخدم إعادة استخدام الشيفرة المصدرية لبناء نسخٍ معدّلةٍ أو مطورةٍ من البرنامج الأصلي أو حتى برمجيات جديدة بالكامل دون الحاجة إلى إعادة كتابة هذه البرمجيات من الصفر.

كما يؤدّي توزيع الشيفرة المصدرية للمنتج البرمجي إلى زيادة قيمة هذا المنتج بالنسبة للمستخدم بطرقٍ غير مباشرة:

- بإمكان المؤلفين والمدربين الذين لا تربطهم علاقات مع الشركة المطورة للبرنامج تأليف كتبٍ وموادٍ تعليميةٍ أكثر تكاملاً ودقّةً لهذا البرنامج دون الحاجة إلى الإعتماد على الوثائق الأصلية التي تصدرها الشركة، مما يجعل التدرب على هذا البرنامج أكثر سهولةً ويشجّع الزبائن على تدريب مستخدميهم ومطوريهم.
- بإمكان الإستشاريين التعرّف على التقنيات المستخدمة في هذه البرمجيات بشكلٍ أفضل ليصبحوا أكثر مهارةً في تصميم وبناء الحلول المعتمدة عليها. سيؤدّي ذلك إلى بناء قاعدةٍ عريضةٍ من المهارات القادرة على بناء الحلول المعقدّة باستخدام هذه البرمجيات، والذي يعني بالتالي تخفيض تكاليف هذه الحلول بالنسبة للمستخدم (نتيجة تنافس الجهات التي تملك المهارات اللازمة لبناء هذه الحلول).
- بإمكان أي جهةٍ إكتشاف المشاكل البرمجية والثغرات الأمنية الموجودة في البرمجيات وتوفير الحلول الملائمة لها، هذه الحلول ستعاد إلى البرنامج الأصلي، مما يسهم في جعل هذا البرنامج أكثر ثباتاً وأمناً وبالتالي يزيد من قيمة هذا البرنامج عبر تقليص تكاليف تشغيله وصيانته.

- بإمكان شركات تطوير البرمجيات الأخرى استخدام هذه الشيفرة المصدرية (في حال تم السماح لها بذلك) لبناء تعديلاتٍ جديدةٍ وإضافاتٍ إلى البرنامج الأصلي لتوفير خدماتٍ لم تكن موجودة ضمنه بالأساس. ويستطيع المستخدمون استثمار هذه التعديلات لبناء أنظمةٍ أكثر فاعليةً، وهو ما يعني زيادة القيمة الإجمالية للبرنامج الأصلي.

وتتجلى القيمة المضافة بشكلٍ واضحٍ في البرمجيات التي توزع عادةً مع شيفرتها المصدرية، كنظام التشغيل لينكس ومخدّم الوب أباتشي، وهي تجعل هذه البرمجيات منتجاتٍ متكاملةً (أي المنتجات المرفقة مع جميع الملحقات الضرورية لضمان تطابق القيمة التي سيحصل عليها المستخدم مع القيمة التي وعد بها المنتج). تتضمن هذه الملحقات الدعم الفنّي، تعليمات الإستخدام، خدمات التدريب، الخدمات الإستشارية وغيرها.

إذا كنت تواجه منافسة الشركات العملاقة فإنك تعرف دون شكّ عما نتحدّث، فهذه الشركات أكثر قدرةً على بناء منتجاتٍ متكاملةٍ نظراً لحجم الموارد الهائل المتاح لهم. مع مرور الوقت، ولكي تبقى قادراً على منافسة هذه الشركات لا بدّ لك من البحث عن أساليب لتوفير نفس القيمة التي يقدّمها منافسوك ضمن منتجاتهم، وإذا أخذنا بعين الإعتبار الفارق في الموارد المتاحة لشركتك ولهذه الشركات العملاقة فإن هذه الأساليب لا يجب أن تتطلب القيام بجميع المهام ضمن الشركة فقط.

وبما أن الهدف الأساسي لشركتك هو الربح المادّي، فزيادة قيمة المنتجات لن تعني شيئاً ما لم تنعكس بشكلٍ إيجابي على إيرادات وأرباح الشركة.

قبل أن نبحث في بعض الأساليب الممكنة لتحقيق ذلك باستخدام البرمجيات مفتوحة المصدر، لا بدّ من استعراض الأسلوب التقليدي المستخدم من قبل معظم شركات تطوير البرمجيات وإيضاح ضرورة تغيير هذا الأسلوب في حالة المصادر المفتوحة.

آلية الربح التقليدية في صناعة البرمجيات

تبنى آلية الربح المستخدمة حالياً في صناعة البرمجيات التجارية على بيع المستخدم (الحقّ) في استخدام البرنامج وليس بيعه الملكية الفعليّة لهذا البرنامج (كتملكه لمنتجٍ فيزيائيٍ ملموس)، وحتى عندما يشتري المستخدم برنامجاً من أحد المتاجر على شكل قرص مدمج مثلاً فإن ما يدفع مقابله في الحقيقة هو الحقّ في استخدام هذا البرنامج وليس ملكية هذا البرنامج.

تعتمد هذه الآلية على الأساس القانوني الذي يعتبر الشيفرة المصدرية المستخدمة لبناء هذا المنتج تندرج تضمن الملكية الفكرية للمطور، وهي محمية بقوانين حماية الملكية الفكرية، ولذلك بإمكان صاحب هذه الملكية الفكرية التحكّم بأسلوب توزيعها واستخدامها عبر عقودٍ قانونيةٍ كإتفاقيات الترخيص المرافقة للبرمجيات، وبإمكانه أيضاً الحصول على مقابلٍ مادّيٍ من الأفراد والمؤسسات التي ترغب في إبرام هذه العقود.

تتمتع تراخيص حقّ الإستخدام من وجهة النظر التجارية البحتة بميزاتٍ عدّة وخصوصاً بالنسبة للشركات المنتجة للبرمجيات، وأحياناً بالنسبة للمستخدم لهذه البرمجيات. يمكن مثلاً تصميم هذه التراخيص لتلبّي المتطلبات المختلفة، كالترخيص حسب عدد المستخدمين، أو حسب عدد المعالجات، أو للمؤسسة بأكملها أو حتّى لجزءٍ من هذه المؤسسة. كما يمكن

تطبيق تراخيص الإستخدام تبعاً للإستخدام الفعلي للبرمجيات، يمكن مثلاً ترخيص برنامج يؤدّي وظيفةً معينةً بشكل مختلف عن ترخيص نفس البرنامج عند أدائه وظيفةً مختلفةً، هذا الأسلوب يتيح للشركة المنتجة تسعير برمجياتها بشكل مرن يلبّي متطلبات الزبائن المختلفة (قد تتيح الشركة المنتجة استخدام برمجياتها مجاناً للإستخدامات غير التجارية مثلاً، كما يمكن أن تحدد سعراً أقل للبرمجيات المستخدمة داخل المؤسسة فقط من تلك المستخدمة لتقديم خدماتٍ خارج المؤسسة).

كما أن رسوم تراخيص استخدام البرمجيات لا تتضمن تكاليف الخدمات الإضافية كالدعم الفنّي والخدمات الإستشارية. أي أنّ رسوم تراخيص الإستخدام هي الإيرادات الفورية التي تحصل عليها الشركة المنتجة عند شحن المنتج مباشرةً، أما رسوم الخدمات فهي إيرادات غير فورية تحصل عليها الشركة مع مرور الوقت عند تقديم هذه الخدمات للزبائن. هذا المفهوم هام جداً بشكلٍ خاص لحالة الشركات الصغيرة الجديدة والتي تحتاج إلى زيادة إيراداتها بسرعةٍ لإرضاء المستثمرين وتوفير المبالغ اللازمة للتوسع.

وبما أنّ رسوم الترخيص غير مرتبطة مباشرةً بجهود الأيدي العاملة فإنه من الممكن التفاوض حول هذه الرسوم بين الشركة المنتجة والزبون بناءً على القيمة التي ستقدّمها هذه البرمجيات للزبون (فقد تحتسب هذه القيمة كنسبةٍ من المبالغ التي سيوفرها إستخدام هذه البرمجيات على الزبون، فإذا كانت هذه المبالغ كبيرةً ستكون قيمة البرمجيات عاليةً). كما قد يكون سعر برنامجٍ آخر أقلّ بكثير نتيجةً لسهولة إعادة إنتاج هذا البرنامج. إذا اعتبرنا أن الزبائن وظروف الطلب في السوق قد حددت قيمةً مرتفعةً لمنتجٍ برمجيٍ ما (ومع تساوي جميع العوامل الأخرى) فإن الأرباح الصافية التي ستجنيها الشركة المنتجة لهذا البرنامج

ستكون أكبر من أرباحها الصافية التي ستجنيها من الخدمات، وهو ما سينعكس مباشرةً على زيادة أرباح هذه الشركة.

وتنطوي تراخيص حقوق إستخدام البرمجيات على العديد من الجوانب السلبية أيضاً، فقد تصبح إجراءات الترخيص معقدةً للغاية بالنسبة لكل من الشركات المنتجة وزبائنها. وقد تتسبب الإيرادات الفورية الناتجة عن رسوم ترخيص إستخدام البرمجيات في نشوء طفراتٍ غير مرغوبةٍ في أرباح الشركة، خاصةً عندما يكون عدد الزبائن محدوداً وحصة كل زبون من الإيرادات مرتفعةً. كما أنّ هوامش الربح الكبيرة في تراخيص إستخدام البرمجيات قد تتعرض لضغوطٍ من قبل المنافسين القادرين على توفير البرمجيات البديلة بأسعار مخفّضة أو حتى توزيعها مجاناً بشكلٍ مستقل أو ضمن برمجياتٍ أخرى.

لسنا هنا بصدد إغداق المديح على آلية الربح التقليدية المستخدمة حالياً في صناعة البرمجيات ولا بذمّها على اعتبار أنها يجب أن تستبدل ببدائل أكثر فاعليةً، ولكنّ ما نريد التركيز عليه هو النقاط التالية:

- لا بدّ للشركات التي تفكّر في تطبيق إستراتيجية تطويرٍ باستخدام المصادر المفتوحة أن تقوم بتقييم المصادر المفتوحة من وجهة نظرٍ تجاريةٍ، دون الإنقياد خلف شعاراتٍ برّاق. ينبغي أن تدرك هذه الشركات بوضوحٍ ميزات وعيوب المصادر المفتوحة من وجهة نظرٍ تجاريةٍ بحتة.
- ترتبط آلية جني الأرباح في صناعة البرمجيات إرتباطاً وثيقاً بكيفية ترخيص هذه البرمجيات، سواء كانت الإستراتيجية المتبعة في التطوير هي المصادر المفتوحة أو المغلقة.

- ينبغي أن تقوم أية شركة بتحديد أسعارٍ لمنتجاتها وخدماتها، حتى في حالة المصادر المفتوحة (وقد يكون هذا السعر صفراً لبعض المنتجات أو الخدمات) . وتعتبر الشركة أكثر نجاحاً من الناحية المالية إذا كانت قادرةً على تحديد أسعار منتجاتها وخدماتها تبعاً للقيمة السوقية لها وليس تبعاً تكاليف الإنتاج. لذلك ينبغي أن نبحث في الخيارات المتعددة لآليات الربح باستخدام المصادر المفتوحة لتحديد المواقع التي يمكن فيها تسعير المنتجات والخدمات تبعاً لقيمتها السوقية عوضاً عن تسعيرها تبعاً لكلفة إنتاجها.

آليات الربح في المصادر المفتوحة

سنحاول هنا الإجابة على التساؤل الشائع هذه الأيام عند طرح فكرة التحوّل إلى المصادر المفتوحة في شركات تطوير البرمجيات:
كيف ستتمكن شركات تطوير البرمجيات من تحويل منتجها البرمجي إلى برنامج مفتوح المصدر واستغلال القيمة المضافة الناتجة عن ذلك بالنسبة للزبائن وترجمتها إلى إيراداتٍ وأرباحٍ للشركة؟ أو بمعنى آخر: كيف يمكن أن تجني شركات تطوير البرمجيات الأرباح باستخدام المصادر المفتوحة؟

ذكرنا فيما سبق أنّ شركات تطوير البرمجيات لن تجني أرباحها في المصادر المفتوحة من بيع تراخيص إستخدام البرمجيات التقليدية، وهو ما يشكل مثالاً حيوياً لوجهة نظر الكثيرين من الصناعيين الذين يعتبرون أنّ الملكية الفكرية يجب ان تكون "حرّةً" بمعني أنّ بإمكان أي شخصٍ نسخها واستخدامها دون مقابل. (وقد يعتبر البعض أنّ البرمجيات مفتوحة المصدر لا تندرج تحت إطار الملكية الفكرية على الإطلاق كبعض أنصار المصادر المفتوحة). ويرى

بعض المحللين أنّ الملكية الفكرية من حيث (محتواها) ستصبح مجانيةً أو شبه مجانيةٍ في المستقبل القريب، ويقترحون البحث عن آلياتٍ جديدةً لاستغلال هذا المحتوى مادّياً مبنية أساساً على توفير الخدمات المتعلقة بهذا المحتوى (كالخدمات الإستشارية).

لقد تمت تجربة العديد من آليات الربح في عالم المصادر المفتوحة، وهي بمجملها تفترض إلغاء رسوم تراخيص إستخدام البرمجيات، سنحاول فيما يلي استعراض هذه الآليات بمزيد من التفصيل، وسنبحث أيضاً:

- شركات تطوير البرمجيات التي تستخدم كلاً من هذه الآليات.
- تراخيص المصادر المفتوحة الأكثر ملاءمةً لكل آليةٍ على حدة.
- فرص التميز المتاحة للشركات التي ستطبق هذه الآليات.
- إمكانية تحديد أسعار ومنتجات الشركة بناءً على القيمة السوقية لها وليست تكلفة إنتاجها.

مزودوا الخدمة

في هذا النموذج من آليات الربح في عالم المصادر المفتوحة تحصل الشركات على إيراداتها عبر توزيع البرمجيات، خدمات التدريب، الخدمات الإستشارية، خدمات التطوير الخاصة، وخدمات الدعم الفني عوضاً عن تحصيل الأرباح من بيع تراخيص إستخدام البرمجيات. هذا النموذج هو أول الآليات التجارية في عالم المصادر المفتوحة والذي اقترحه ريتشارد ستولمان عند طرحه لمفاهيم المصادر المفتوحة، وهو أكثر النماذج استخداماً من قبل الشركات العاملة في صناعة البرمجيات مفتوحة المصدر، كريد هات RedHat و سوزي

SuSE. تستخدم هذه الشركات عادةً ترخيص GPL ولكنها تستطيع استخدام أيٍ من التراخيص الأخرى عند الحاجة.

تحصّل الشركات إيراداتها في هذا النموذج من بيع منتجين أساسيين: المنتجات الفيزيائية كالأقراص المدمجة وتعليمات الإستخدام المطبوعة، والخدمات كخدمات الدعم الفنّي.

بإمكان الشركات التي تطبق هذا النموذج في عملها التميز عبر تقديم توزيعات برمجية متكاملةٍ وأكثر سهولة في التثبيت والإستخدام، إضافةً إلى نوعية وأسعار الخدمات التي تقدّمها.

يصعب في هذا النموذج تحديد الأسعار تبعاً لقيمة المنتج بالنسبة للزبون، وتحدّد الأسعار عادةً بناءً على تكاليف تقديم المنتجات والخدمات نظراً للتنافس مع الشركات الأخرى ومحدودية الأسعار التي سيدفعها الزبون لقاء الحصول على هذه المنتجات والخدمات. ولكن السمعة الطيبة للشركة قد تبرر إرتفاع أسعار منتجاتها وخدماتها إلى حدٍ ما.

المنتجات الداعمة

تشكّل البرمجيات مفتوحة المصدر في هذا النموذج منتجاً داعماً لمنتجٍ برمجيٍ آخر غالباً ما يكون مغلق المصدر. لن تحصل الشركة على أي مقابلٍ لقاء توزيع البرنامج مفتوح المصدر (أو ربما قد تحصل على النزر اليسير من العوائد)، ولكن توفير هذا المنتج قد يدفع المستخدمين إلى شراء برمجياتٍ أخرى تباع وفق آلية الربح التقليدية (أي مقابل رسوم

تراخيص الإستخدام). يستخدم هذا النموذج من قبل شركة Netscape وشركة Sendmail Inc.

إذا احتوى البرنامج مفتوح المصدر على ملفاتٍ مصدريةٍ مشتركةٍ مع البرامج المغلقة التي تطورها الشركة لا بدّ من اختيار ترخيص إتفاقية ترخيص للمصادر المفتوحة تتيح توزيع هذه الشيفرة المصدرية مع البرنامج مفتوح المصدر مع الحفاظ على حقّ الشركة في استخدام هذه الشيفرة المصدرية في برمجياتها المغلقة التي توزّع ضمن إتفاقيات ترخيص الإستخدام التقليدية. ولذلك يجب أن تتجنّب الشركة استخدام إتفاقية ترخيص GPL والتي لا تنسجم مع هذه المتطلبات، وقد تعتبر إتفاقيّة ترخيص BSD أو Mozilla Public License الخيار الملائم (وقد تقوم الشركة بتطوير إتفاقية ترخيصٍ جديدةٍ تلائم متطلباتها).

إذا أرادت الشركة تحقيق بعض الإيرادات من توزيع البرمجيات مفتوحة المصدر ضمن هذا النموذج (أي إذا لم تشأ هذه الشركة توزيع هذه البرمجيات دون مقابل) فإن آلية تحصيل الإيرادات تشابه تلك المطبّقة في نموذج مزودي الخدمة.، ولكن الحصة الأكبر من إيرادات الشركة ستنتج عن بيع البرمجيات المغلقة. سيساعد توزيع البرمجيات مفتوحة المصدر في دفع الزبائن إلى شراء البرمجيات المغلقة لأسبابٍ عدّة:

- تحسين سمعة الشركة وزيادة شهرتها.
- زيادة فاعلية وأداء البرمجيات المغلقة (أي إضافة المزيد من القيمة إليها).
- توسيع قاعدة المطورين والمستخدمين الماهرين باستخدام جميع منتجات الشركة.

بإمكان الشركات التي ستطبّق هذا النموذج التميز عبر جودة البرمجيات المغلقة التي تنتجها، كما تستطيع تسعير منتجاتها تبعاً لقيمتها بالنسبة إلى الزبائن.

المنتجات المكمّلة

يستخدم هذا النموذج بشكل أساسي من قبل الشركات المنتجة للتجهيزات والتي تستخدم المصادر المفتوحة لتوفير برمجيات مكمّلة لمنتجاتها (كبرامج القيادة مثلاً) توزّع مجاناً مع هذه التجهيزات. وتشمل التجهيزات جميع أشكال التجهيزات الحاسوبية بدءاً من الشرائح الصغرية وانتهاءاً بأنظمة الحواسيب المتكاملة. أما البرمجيات المكمّلة فتشتمل على برامج القيادة وأدوات التحكم والتطبيقات البرمجية وأنظمة التشغيل.

من الشركات التي تعتمد هذا النموذج في أعمالها شركة VALinux والتي تبيع مخدّماتٍ وحواسيب معدّة مسبقاً باستخدام نظام التشغيل لينكس.

تحصل الشركات التي تستخدم هذا النموذج على إيراداتها من عوائد بيع التجهيزات، كما أنّ استخدام البرمجيات مفتوحة المصدر كبرمجياتٍ داعمةٍ للتجهيزات أيضاً سيزيد من مبيعات هذه التجهيزات عبر توسيع قاعدة المطورين المهرة باستخدام هذه التجهيزات والذين بدورهم سيضيفون المزيد من القيمة على هذه المنتجات بالنسبة إلى الزبائن.

بإمكان الشركات التي ستطبّق هذا النموذج أن تتميز عبر ميزات ومرونة وأداء وكلفة التجهيزات التي تنتجها، ودور البرمجيات مفتوحة المصدر يتجلى في تدعيم هذه التجهيزات لرفع مستوى أدائها وزيادة مرونتها وتخفيض كلفتها. وبما أن الشركة تعتمد على بيع

منتجاتٍ عرضةٍ للمنافسة من قبل الشركات الأخرى المنتجة للتجهيزات فإن التسعير في هذا النموذج يعتمد عادةً على تكاليف الإنتاج.

السلع المضافة

يستخدم هذا النموذج من قبل الشركات التي تقوم بإنتاج وتوزيع الكتب والمنتجات الأخرى (مقارنةً بالبرمجيات والخدمات) المتعلقة بالبرمجيات مفتوحة المصدر. لا تقوم الشركات في هذا النموذج بالمساهمة في تطوير أيٍ من البرمجيات مفتوحة المصدر، بل تعتمد على برمجيات تم تطويرها من قبل الآخرين.

من الأمثلة عن الشركات التي تعتمد هذا النموذج شركة Oreilly & Associates والتي تقوم بإصدار الكتب والمنشورات المتعلقة بالبرمجيات مفتوحة المصدر. يحتاج هذا النموذج إلى إتفاقية ترخيصٍ تتيح إرفاق البرمجيات مفتوحة المصدر مع المنتجات المباعة (كالكتب مثلاً) وهو ما توفره جميع إتفاقيات ترخيص المصادر المفتوحة.

يمكن التميز ضمن هذا النموذج عبر تحسين نوعية السلع المباعة بحدّ ذاتها. وبما أنّ المنتجات المباعة تندرج أكثر ضمن المنتجات الفيزيائية أكثر منها ضمن النتاج الفكري فإن التسعير عادةّ ما يبنى على تكاليف الإنتاج أكثر منها على قيمة هذه المنتجات بالنسبة للزبون، وهنا أيضاٍ قد تبرر السمعة الطيبة للشركة إرتفاع أسعار منتجاتها.

دعم الخدمات المدفوعة

في هذا النموذج تستخدم الشركات البرمجيات مفتوحة المصدر التي تطورها وتوزعها مجاناً لدفع الزبائن إلى الإشتراك بخدمات تعود بالربح على هذه الشركة (قد تعود هذه الخدمات بالأرباح على الشركة نتيجة رسوم الإشتراك أو خدمات الإعلانات) .

من الأمثلة على هذا النموذج شركة تنتج برنامجاً مفتوح المصدر وتوزّعه ليمكّن الزبائن من الإشتراك بخدمة الألعاب التي توفرها هذه الشركة عبر الإنترنت.

يجب أن تختار الشركات التي ستطبّق هذا النموذج إتفاقية ترخيصٍ تحول دون استخدام الشركات المنافسة لشيفرتها المصدرية وتحويلها إلى برامج خاصةٍ بهم. وقد تلبّي إتفاقية ترخيص GPL أو Mozilla Public License هذه المتطلبات.

تتميز الشركات التي تطبّق هذا النموذج عبر مستوى ونوعية الخدمات التي تقدّمها، وقد تلعب نوعية ومزايا البرنامج مفتوح المصدر الذي تقوم الشركة بتوزيعه دوراً بسيطاً في ذلك، ولكنّ الأساس هو الخدمات المقدّمة من قبل هذه الشركة. ويعتبر توفير الخدمات المميزة والمفيدة للمستخدم والتي يصعب توفيرها من قبل المنافسين أمراً أساسياً لتبرير تسعير هذه الخدمات تبعاً لقيمتها بالنسبة للمستخدم (فقد تسعر شركة تخديم الألعاب عبر الإنترنت خدماتها بناءً على القيمة الترفيهية التي تقدّمها للمستخدم مقارنةً مع البدائل الأخرى كألعاب الفيديو المنزلية أو القنوات التلفزيونية).

البيع أولاً ومن ثمّ التحرير

ويعتبر هذا النموذج مطابقاً لنموذج المنتجات الداعمة مع تكراره وتطويره بمرور الزمن. تبني الشركات التي تستخدم هذا النموذج آليات تطوير وترخيص البرمجيات التي تنتجها بشكلٍ يتيح إصدار هذه البرمجيات في البداية كبرمجياتٍ مغلقةٍ تقليدية ومن ثمّ تحويلها إلى برمجياتٍ مفتوحة المصدر عند بلوغها حداً معيناً في دورة حياتها تصبح معه ميزات تطوير هذه البرمجيات في بيئة مفتوحة المصدر أكثر جدوى من الإيرادات التي تنتج عن بيع تراخيص إستخدام هذه البرمجيات. ستضيف المنتجات التي تمّ تحريرها المزيد من القيمة المضافة إلى البرمجيات المغلقة الأخرى كما في نموذج المنتجات الداعمة، وعند استخدام إتفاقية الترخيص الملائمة، يمكن بناء البرمجيات الجديدة المغلقة جزئياً على الشيفرة المصدرية للمنتجات السابقة التي أصبحت مفتوحة المصدر.

يواجه التطبيق العملي لهذا النموذج مشكلتين لا بدّ من تجاوزهما: المشكلة الأولى هي تحديد التوقيت الملائم في دورة حيات المنتج لتحويله إلى المصادر المفتوحة، فالقيام بذلك في وقتٍ مبكر قد يعني ضياع بعض الإيرادات الناتجة عن رسوم تراخيص الإستخدام، في حين يعني التأخر في ذلك تبدد إهتمام مطوري المصادر المفتوحة في المساهمة في تطوير هذا البرنامج. والمشكلة الثانية متعلقة بالأولى: طالما أنّ الزبائن يعلمون مسبقاً أنّ هذا البرنامج سيتحول إلى برنامج مفتوح المصدر في مرحلةٍ ما، فإنّ عملية إقناعهم بشراء هذا البرنامج ستصبح أكثر صعوبةً.

يدفع الزبائن الذين يقدمون على شراء البرنامج عند إطلاقه مبالغ إضافيةً لقاء الحصول على هذا البرنامج في وقت مبكر، ويمكن أن تعكس السياسة التسعيرية للشركة هذه النقطة. وبعد أن يتحول البرنامج إلى المصادر المفتوحة فإنه يصبح مجرد سلعة، ولكنه يبقى قادراً على

إضافة المزيد من القيمة إلى البرمجيات المغلقة الجديدة والتي قد تعكس أسعار تراخيص إستخدامها هذه القيمة المضافة.

ترخيص العلامة التجارية

تقوم الشركات التي تطبّق هذا النموذج بتوزيع البرنامج نفسه كبرنامجٍ مفتوح المصدر، ولكنها تحتفظ بحقوق العلامة التجارية وحقوق الملكية الفكرية لهذا المنتج، وتتلقّى رسوماً من الشركات الأخرى التي ترغب في استخدام هذه العلامات التجارية لبناء منتجاتٍ معدّلةٍ وتوزيعها تحت العلامة التجارية نفسها. تتطلب هذه العملية توفّر المنتج بصيغتين مختلفتين تحمل كل منهما إسماً مختلفاً: المنتج "الرسمي" والذي يحمل العلامة التجارية، والمنتج "غير الرسمي" والذي يحمل إسماً مختلفاً.

لا يمكن إستخدام هذا النموذج مع البرمجيات الحرّة التقليدية لأن هذه العلامات التجارية لهذه البرمجيات لم تكن مسجلةً بشكلٍ رسميٍ كعلاماتٍ تجاريةٍ مسجّلة. أما البرمجيات التجارية التي ستتحول إلى برمجيات مفتوحة المصدر فهي غالباً ما تترافق مع علاماتٍ تجاريةٍ وشعاراتٍ مسجلةٍ والتي ستبقى تحت سيطرة الشركة المنتجة لهذه البرمجيات. (على سبيل المثال: مع أنّ شركة Netscape قد حولت برنامجها Netscape Communicator إلى برنامجٍ مفتوح المصدر، فإن شركة Netscape وحدها تملك الحقّ في استخدام هذه الشيفرة المصدرية لبناء منتجٍ يسمّى Netscape Communicator، وأية برمجياتٍ تبنى باستخدام هذه الشيفرة المصدرية من قبل الآخرين يجب أن تحمل أسماءً مختلفةً).

بشكلٍ عام يتم بناء النسختين المختلفتين من البرنامج (النسخة التي تحمل العلامة التجارية والنسخة التي لا تحمل هذه العلامة) باستخدام نفس الشيفرة المصدرية، ومع ذلك فقد تحمل كل من هاتين النسختين قيمةً سوقيةً مختلفة. (فقد تمر النسخة التي تحمل العلامة التجارية عبر المزيد من مراحل التجربة مثلاً).

إذا قات الشركة بتحويل برنامجٍ ما إلى برنامجٍ مفتوح المصدر فقد تستخدم جهات أخرى هذا البرنامج أو شيفرته المصدرية وتدفع للشركة مقابل الحصول على ترخيصٍ لاستخدام العلامة التجارية مع هذا المنتج. يمكن أن تعكس هذه المبالغ قيمة العلامة التجارية وسمعة الشركة التي تملكها في السوق.

شركاء الأعمال

يشكّل هذا النموذج دمجاً للعديد من النماذج التي استعرضناها حتّى الآن، وبشكل خاص نموذجي ترخيص العلامة التجارية ودعم الخدمات المدفوعة. إن المواد الأولية المتوفرة في نموذج دعم الخدمات المدفوعة متاحة للجميع، ولكن بعض الشركات قد تقوم بعملها بشكلٍ أفضل من غيرها مما يؤدّي إلى بناء علامةٍ تجاريةً مميزةٍ مرتبطةٍ بالخدمات التي تقدّمها هذه الشركة. وفي حال رغبت هذه الشركة في توسيع أعمالها فإن بمقدورها أن تقوم بذلك عبر بناء شبكةٍ من شركاء الأعمال وذلك بالترخيص لمطورين آخرين باستخدام العلامات التجارية الخاصّة بالشركة لإنشاء مؤسساتٍ جديدةٍ مرتبطةٍ بالشركة الأم لتوفير خدمات الدعم الفنّي وتطوير البرمجيات في مناطق جغرافيّةٍ معينةٍ أو أسواق محددة.

لا تقوم الشركات الأساسية في هذا النموذج بإتاحة استخدام علاماتها التجارية لشركائها وحسب، بل تقدّم لهم الخدمات التدريبية والخدمات المختلفة (كخدمات التسويق والإعلان). وتحصل في المقابل على إيراداتها من عوائد عقود الشراكة والحصول على نسبةٍ من أرباح هذه الشراكات.

كيف تستطيع المؤسسات توفير النفقات باستخدام برمجيات المصادر المفتوحة؟

يعتمد الخيار بين استخدام البرمجيات ذات المصادر المفتوحة والبرمجيات المغلقة على عدّة عوامل منها: توفّر التطبيقات اللازمة، توفّر المهارات المطلوبة لتشغيل وإدارة هذه البرمجيات وتوفّر الدعم الفنّي لها، إلا أنّ أهمّ هذه العوامل هو التكلفة المادّية لهذه البرمجيات.

إن اختيار البرمجيات مفتوحة المصدر لبناء حلول تقنية المعلومات بالإضافة إلى تقديم مستوياتٍ مميزةٍ من الأداء والوثوقية سيوفّر على المؤسسة الكثير من التكاليف والنفقات، وسنبحث فيما يلي في المزايا المترافقة مع استخدام البرمجيات مفتوحة المصدر من الناحية الإقتصادية.

تركّز معظم الدراسات التي تجرى لتحديد تكاليف التشغيل الكليّة لأنظمة تقنيّة المعلومات على المهام الأساسية كتكاليف شراء التجهيزات والبرمجيات، تكاليف التركيب والإعداد، تكاليف الإدارة والصيانة والتشغيل وتكاليف التدريب، متجاهلةً بعض الفروقات الأساسية بين البرمجيات المختلفة، كإختلاف أساليب ترخيص هذه البرمجيات ومستوى أدائها ووثوقيتها...

ينطوي إعتماد البرمجيات مفتوحة المصدر على وفرٍ في تكاليف التشغيل الكليّة لعدة أسباب:

1. بإمكان البرمجيات مفتوحة المصدر تشغيل عدّة تطبيقات (وبالتالي تقديم عدّة خدمات) ضمن مخدّمٍ واحد، مما يلغي الحاجة إلى عددٍ كبيرٍ من المخدّمات.
2. يفوق أداء البرمجيات مفتوحة المصدر عادةً أداء البرمجيات المغلقة، وبإمكان هذه البرمجيات القيام بنفس المهام ضمن تجهيزات أقل كلفةً من تلك التي تتطلبها نظيرتها المغلقة.
3. إن تكلفة الحصول على البرمجيات مفتوحة المصدر أقل من تكلفة البرمجيات المغلقة.
4. إن تكاليف إدارة وتشغيل برمجيات المصادر المفتوحة أقل من تلك اللازمة للبرمجيات المغلقة.

تجبر البرمجيات المغلقة مستخدميها على استثمار الكثير من الوقت والمال في مهام لا تشكل أية أهميةٍ بالنسبة إليهم، ولذلك فإن اعتماد البرمجيات مفتوحة المصدر سيوفّر على المستخدمين هذا الإستثمار من الوقت والمال عبر إلغاء الحاجة إلى:

- إدارة ومتابعة تراخيص الإستخدام: إن القيود القانونية المعقدّة التي تفرضها إتفاقيات ترخيص استخدام البرمجيات المغلقة تتطلب من مستخدم هذه البرمجيات توخي الحيطة والحذر في كل ما يتعلّق بهذه التراخيص خشية الوقوع في مشاكل قانونية نتيجة خرق هذه الإتفاقيات. أي أن المستخدم مطالب بالتحقق من أن جميع البرمجيات المستخدمة في مؤسسته مرفقة بتراخيص الإستخدام الضرورية بشكلٍ دائم.
- تفسير إتفاقيات الترخيص: لقد وقعت العديد من الشركات في مشاكل مع شركات تطوير البرمجيات التجارية نتيجة الفهم الخاطئ لاتفاقيات

الترخيص والذي أدّى إلى حصول الشركة المستخدمة للبرمجيات على الترخيص الخاطئ أو العدد الخاطئ من التراخيص (وفي بعض الحالات كان المسؤول عن هذا الخطأ هو موظّف المبيعات في الشركة المنتجة ذاتها!).

- مسلسل التطويرات الدائمة: تستخدم بعض شركات تطوير البرمجيات المغلقة التطويرات التي تجريها على منتجاتها لدفعك إلى شراء برمجياتٍ لست بحاجةٍ إليها.

في المقابل لا تتطلب البرمجيات مفتوحة المصدر شراء برمجياتٍ جديدةٍ عند إصدار تطويراتٍ أو تعديلاتٍ جديدة.

- ترخيص البرمجيات لكل مستخدم: وهو ما يدعى Client Access License، والذي أعتبره بمثابة مطبعة للأوراق النقدية لشركات تطوير البرمجيات المغلقة، فأنت تدفع لقاء استخدامك لبرمجياتهم على مخدّماتك، وتدفع لقاء استخدامك لبرمجياتهم على حاسبك الشخصي أيضاً، ومن ثمّ تجبر على أن تدفع لقاء تمكين هذه البرمجيات مع التخاطب مع بعضها البعض!

إنّ استخدام البرمجيات مفتوحة المصدر سيوفّر عليك هذه المبالغ الطائلة لاستخدامها في مواقع ضروريةٍ أخرى في أعمالك.

- عدد المخدّمات: تملي نصائح شركات تطوير البرمجيات المغلقة استخدام مخدّمٍ مخصصٍ لكل خدمةٍ تريد توفيرها ضمن شبكتك، وكل مخدّمٍ إضافي سيكلّفك ثمنه وتكاليف تراخيص البرمجيات التي ستعمل ضمنه وتكاليف إدارته وتشغيله، وهي تكاليف قد تصل لأرقامٍ فلكيةٍ في بعض الأحيان.

بإمكان البرمجيات مفتوحة المصدر تشغيل عدّة تطبيقات (وبالتالي تقديم عدّة خدمات) ضمن مخدّمٍ واحد.

- عدد فنيي إدارة وتشغيل الأنظمة: إن وجود عددٍ أقل من المخدّمات سيتطلب عدداً أقل من من الفنيين لإدارة هذه المخدّمات وتشغيلها.
  كما تتميز برمجيات المصادر المفتوحة بمستوىً عالٍ من الوثوقيةٍ والثباتٍ يقلّص متطلبات الصيانة والتشغيل.
- الفيروسات: هناك ما يربو على 100000 فيروساً لنظام التشغيل ويندوز، في حين يوجد أقل من 100 فيروسٍ لنظام التشغيل مفتوح المصدر لينكس.
  إنّ متطلبات مكافحة الفيروسات في البرمجيات المغلقة تشكّل جزءاً هاماً من تكاليف تشغيل وصيانة هذه البرمجيات.
- الوثوقية: يتسبب توقّف البرمجيات عن العمل لأي سببٍ كان في خسارةٍ كبيرةٍ للمؤسسة، خصوصاً إذا كانت هذه البرمجيات تشكّل عصباً أساسياً لأعمال هذه المؤسسة، وهو حال معظم المؤسسات والشركات في عصر الإنترنت والتجارة الإلكترونية.
  تتفوق البرمجيات مفتوحة المصدر بمراحل على نظيرتها المغلقة من حيث الوثوقية والإعتمادية، وهو ما يعني ضمان إستمرارية العمل والتخلّص من مشاكل توقّف هذه البرمجيات عن العمل.

يشكّل خيار التحوّل إلى البرمجيات مفتوحة المصدر بالنسبة إلى المؤسسات والجهات الحكومية والشركات التجارية في الوطن العربي قراراً إستراتيجياً لا بدّ من البدء بأخذه بعين الإعتبار عند بناء حلول تقنية المعلومات في هذه المؤسسات، فالبرمجيات مفتوحة المصدر

تشكّل فرصةً ذهبيةً لتحويل المبالغ الطائلة التي تدفع كل عامٍ للحصول على تراخيص إستخدام البرمجيات المغلقة إلى شركات تقنية المعلومات العربيّة التي تعمل ضمن نطاق الإقتصاد العربي وتوظّف خبراتٍ عربيةً مما سيساعد على دعم هذه الشركات وتطويرها وإلى بناء المهارات والخبرات العربيّة في مجال تقنية المعلومات.

كما أن المصادر المفتوحة تشكّل حلاً مثالياً للتخلص من القيود التي تفرضها الحكومة الأمريكية على تصدير المنتجات التقنية إلى بعض الدول العربيّة.

أسئلة عليك إجابتها

سواء كنت في موقعٍ يتطلب منك إتخاذ قرارٍ فيما يتعلّق باعتماد البرمجيات مفتوحة المصدر في مؤسستك أو شركتك، أو كنت تبحث عن استراتيجية ملائمة لجني الأرباح من البرمجيات مفتوحة المصدر، عليك أن تقوم بتحديد متطلباتك بدقّة. عليك أن تطرح على نفسك الأسئلة التالية:

- هل أرغب فعلاً في اعتماد استراتيجيةٍ مبنيةٍ على المصادر المفتوحة لجني الأرباح؟
- ما هي المخاطر التي تشكلها ثورة المصادر المفتوحة على أعمالي؟
- ما هي الفرص التي قد أتمكن من استغلالها باعتماد المصادر المفتوحة؟
- ما هي المنتجات التي أقوم بإنتاجها وتوزيعها (سواء كانت منتجاتٍ برمجيةٍ أو خدمية أو غيرها)؟
- ما هي التكاليف التي أدفعها سنوياً لقاء استخدام البرمجيات المغلقة؟

الفصل السادس

البعد الأمني والمصادر المفتوحة

مقدّمة

قد يبدو للوهلة الأولى أنّ استخدام البرمجيات مفتوحة المصدر لحماية المعلومات خيار مشكوك في صحته، فكيف يمكن للشركات والمؤسسات أن تستخدم برمجيات متاحة للجميع (بما فيهم قراصنة المعلومات والمخترقين) لحماية أثمن موجودات هذه الشركة أو المؤسسة من المعلومات؟ وعلى الرغم من ذلك فإن غالبية أنظمة حماية المعلومات كالجدران النارية وأنظمة التشفير والمستخدمة في الكثير من الشركات الكبرى والهيئات الحكومية حول العالم مبنية باستخدام البرمجيات مفتوحة المصدر. فما هو إذاً ما تعرفه هذه الشركات عن البرمجيات مفتوحة المصدر والذي لا تريدك شركات البرمجيات التجارية أن تعرفه؟

عملياً، وعندما يتم إعدادها بشكل جيد، تقدّم البرمجيات مفتوحة المصدر حلولاً أفضل وأكثر فاعليةً وأمناً لحماية المعلومات من نظيراتها المغلقة، عدا عن كونها أرخص ثمناً.

البحث عن أمن المعلومات

تنحدر غالبية خبراء الأنظمة المفتوحة من خلفية مرتبطة بشكل ما أو بآخر بنظام التشغيل يونيكس، وقد تعاملوا على الأغلب مع مشاكل الأمن على الإنترنت (ويعود الفضل إلى مجموعة خبراء يونيكس ولينكس في بناء شبكة الإنترنت وتشكيلها بشكلها الحالي). واجهت هذه المجموعة الكثير من مشاكل أمن المعلومات مما أدّى إلى بناء مهارات ضخمة لديها في

هذا المجال، فقد تفهمت مشاكل أمن المعلومات بشكل أفضل، وكانت أول من ابتكر أنظمة كشف التسلل والجدران النارية وتقنيات تقييم أمن المعلومات المستخدمة حالياً. قد تكون برمجيات المصادر المفتوحة عرضةً للإختراق عبر بعض الثغرات الأمنية في بعض الأحيان، ولكن مطوريها أمضوا السنوات العشرين الماضية في العمل على جعل هذه البرمجيات أكثر أمناً من أية برمجيات أخرى.

الأمن والبرمجيات مفتوحة المصدر

ليس من المعيب أن نعترف بأنّ مفهوم الآمن الكامل للمعلومات غير موجود إلا في مخيلتنا، وغاية ما يمكن عمله في سبيل مقاربة هذا المفهوم هو محاولة جعل البرمجيات أو الشبكة آمنةً قدر الإمكان.

لقد قام مطوروا البرمجيات مفتوحة المصدر بالعمل الحثيث لمكافحة المشاكل الأمنية في برمجياتهم، ومن أهم فلسفاتهم الأساسية التي تتعلق بأمن المعلومات: إذا أوليت الثغرات الأمنية في البرمجيات بما يكفي من الرقابة والمتابعة فإنها ستنحسر إلى حدّ الإنعدام. فعندما تكتشف ثغرة أمنية في أحد البرمجيات مفتوحة المصدر سيبادر الكثير من المطورين وخبراء أمن المعلومات (والذين يستطيعون الحصول على الشيفرة المصدرية لهذه البرمجيات) إلى متابعة هذه الثغرة وحلها بسرعة دون الحاجة إلى العودة إلى المطور الأساسي للبرنامج.

كما أنّ معظم المشكلات الأمنية التي تظهر في البرمجيات مفتوحة المصدر هي مجرد مشاكل عامّة تختلف كليّةً عن المشكلات الأمنية الجوهرية التي تظهر في البرمجيات المغلقة. ولا بدّ من الإنتباه إلى هذه النقطة بشكل خاص عند مقارنة المشكلات الأمنية بين

البرمجيات المفتوحة والمغلقة المصدر، فلا بمكن بأي حال من الأحوال أن نعتبر أنّ خمسة مشكلات عامة توازي خمسة مشكلات أمنية جوهرية، أي أن طبيعة الثغرة أو المشكلة الأمنية تلعب دوراً أساسياً في تقييم هذه المشكلة.

عندما تتسبب مشكلة عامة في إعاقة عمل نظام ما فإنها ستكتشف وتحل دون أن يكون لها (بشكل عام) تأثير سلبي يوازي تأثير المشاكل الأمنية الجوهرية، والتي قد تتسبب في نزع جميع وسائل حماية هذا النظام. في أسوأ حالاتها قد تتسبب المشكلات العامة في فقدان المعلومات المخزّنة ضمن النظام أو إيقافه عن العمل، أما المشكلات الأمنية فقد تجعل النظام ساحةً لعبث القراصنة والمخربين، وهو ما قد يتسبب في فقدان المعلومات نتيجة العبث، تحويل المعلومات السرية إلى المنافسين أو الصحافة، تخريب المعلومات المخزّنة في قواعد البيانات أو سرقة معلومات الحسابات المصرفية الخاصة بالزبائن.

والأسوأ من كل ذلك أنّه وبمجرد اكتشاف الثغرات أو المشكلات الأمنية فإن أخبارها ستنتشر انتشار النار في الهشيم، ففي حين يتم اكتشاف المشاكل العامة وحلها، سيتم نشر خبر اكتشاف الثغرات الأمنية حول العالم ليتم استغلال هذه الثغرات في الهجوم على آلاف الأنظمة في كل مكان خلال عدّة ساعات، مما سيجعل الخسائر الناتجة عن هذه الثغرات هائلةً.

عادةً ما يتجنب مدراء الأنظمة استخدام أحدث الإصدارات من البرمجيات حتى تتم تجربتها وتثبت مدى وثوقيتها، ولذلك فإن الأنظمة المستخدمة لن تكون عرضةً لمشاكل عامّة قد تتسبب في إحداث كوارث. أما الثغرات والمشاكل الأمنيّة فقد تظهر في أنظمة مستخدمة على نطاق واسع، مما يزيد أيضاً من الخسائر الناتجة عن هذه الثغرات.

ولأن المشاكل البرمجية العامة تصيب كل أنواع البرمجيات (بما فيها تل كالمغلقة أو مفتوحة المصدر) فإن الفلسفة التي ذكرناها سابقاً بعيدة عن الكمال، ولكننا نعلم أنّ المشاكل الأمنية في البرمجيات مفتوحة المصدر تحل بسرعة عند مستوى الشيفرة المصدرية، وهو ما لا نستطيع ضمانه في البرمجيات التجارية المغلقة، وهو ما قد يعتبر أهم ميزات البرمجيّات مفتوحة المصدر فيما يتعلق بالناحية الأمنيّة، حيث أنّ المشاكل الأمنيّة التي يتم اكتشافها ستحل بسرعة فائقة. في المقابل تتمتع الشركات المنتجة للبرمجيات التجاريّة بسمعة غير محمودة فيما يتعلق بحل المشاكل الأمنية التي تكتشف في برمجياتها، على الرغم من تحسن هذه الناحية مؤخراً، وربما يعود الفضل في ذلك إلى الإعلام وتركيزه على العواقب السلبية الناتجة عن إهمال هذه المشاكل الأمنية.

البرمجيّات مفتوحة المصدر أقلّ كلفةً

تكلّف حلول أمن المعلومات مفتوحة المصدر مبالغ أقلّ بكثير من مقابلتها المغلقة، هذا التوفير في النفقات سيسهم في تدعيم مستويات الأمن في الشركات الصغيرة والمتوسطة والتي لا تملك الكثير من الموارد لإنفاقها على تقنيّة المعلومات ولكنّها بحاجة إلى أمن المعلومات، هذه الشركات عاجزة عادةً عن توفير متطلبات أمن المعلومات نتيجة لتكاليفها الباهظة، وهي غالباً ما تعتمد على ما يكتبه لها القدر فيما يتعلق بأمن معلوماتها، وفي أحسن الأحوال فإنها ستركّز على حماية أهمّ ما تملك من معلومات0 أمّا باستخدام البرمجيّات مفتوحة المصدر فإن هذه الشركات تستطيع حماية كم أكبر من المعلومات دون زيادة الكلفة، كما تتيح إعادة توزيع استثمار الشركة في مواقع أخرى غالباً ما يتم إهمالها والتي قد تكون ذات نفع كبير على المدى البعيد.

يستغرق إصلاح البرمجيات مفتوحة المصدر وقتاً أقل

عندما يتم اكتشاف ثغرة أمنيّة في برنامج ما فإنها ستحل بسرعة أكبر فيما لو كان هذا البرنامج مفتوح المصدر عنها فيما لو كان مغلقاً0 وفيما يختص بأمن المعلومات تعتبر سرعة إيجاد الحلول أمراً فائق الأهميّة.

أظهرت دراسة أجراها موقع SecurityPortal في عام 2000 بأنّ شركة ريدهات RedHat تحتاج وسطياً إلى 11 يوماً لحل المشاكل التي تظهر في نظام التشغيل لينكس، فيما تحتاج مايكروسوفت إلى 16 يوماً، ويستغرق حلّ هذه المشاكل لدى Sun ما يقارب ثلاثة أشهر.

لماذا لا يسهم إغلاق الشيفرة المصدرية في حماية البرمجيات؟

يعتبر البعض أنّ عدم إتاحة الشبفرة المصدرية للبرمجيات يسهم في جعلها أكثر أمناً نتيجة توفر قدر أقل من المعلومات عنها للمخترقين مما يجعل عملية البحث عن الثغرات الأمنية واستغلالها أمراً أكثر صعوبةً. إلا أن ذلك في الواقع غير صحيح، لأنه وعلى الرغم من أهمية توفر الشيفرة المصدرية عند تطوير البرمجيات فإن المخترقين لا يحتاجون عادةً إلى الحصول على الشيفرة المصدرية أثناء بحثهم عن الثغرات الأمنية.

علينا أن نميز في البداية بين النشاطات "التخريبية" والنشاطات "البنّاءة"، فتدمير سيارة على سبيل المثال أسهل بكثير من بناء واحدة جديدة. وفي عالم البرمجيات فإن إيجاد ثغرة أمنية

في برنامج ما واستغلالها أسهل بكثير من تطوير هذا البرنامج وزيادة إمكانياته، وهو ما يمنح المخترقين بعض الميزات في مواجهة مطوري هذا البرنامج، حيث ينبغي على هؤلاء المطورين توخي الحيطة والحذر لكي لا تتضمن الشيفرة المصدرية للبرنامج أية ثغرات أمنية، أما المخترقين فيكفيهم أن يجدوا ثغرةً واحدةً في هذا البرنامج. ويتقاضى مطوروا البرمجيات أجورهم لجعل برمجياتهم تعمل، أما المخترقين فلا حاجة لهم لضمان سير عمل البرمجيات، وكل ما يبحثون عنه هو نقطة ضعف واحدة. وكما سنبين لاحقاً فإن اختراق برنامج ما يتطلب قدراً من المعلومات عن هذا البرنامج أقل بكثير من تلك اللازمة لتعديله.

يبدأ المخترقون عادةً (سواء كان هدفهم برنامجاً مغلقاً أم مفتوح المصدر) بالتعرف إلى المشاكل الأمنية العامة في البرمجيات، ولا حاجة إلى محاولة إخفاء هذه المعلومات، فهي متوفرة وعلى نطاق واسع، ويحتاجها المطورون لحماية برمجياتهم. بعد ذلك يقوم المخترقون باستخدام تقنيات للبحث عن هذه المشاكل، وتقسم هذه التقنيات إلى تقنيات ساكنة وأخرى ديناميكية، ففي الأخيرة يحاول المخترقون البحث عن الثغرات الأمنية عبر تشغيل البرمجيات والعبث بها، أما في التقنيات الساكنة فإنهم يقومون بتفحص الشيفرة المصدرية لهذه البرمجيات.

عندما يستخدم المخترقون التقنيات الديناميكية في البحث عن الثغرات الأمنية فإنهم يقومون بتشغيل البرنامج الهدف ويرسلون إليه معلومات غالباً ما تكون تخريبية لمعاينة رد فعل البرنامج وما إذا كان هذا الرد سيشير إلى ثغرة أمنية ما. وفي هذا السياق لا يوجد أي فرق بين البرمجيات المغلقة أو مفتوحة المصدر لأنّ المخترق لا يقوم بمعاينة الشيفرة المصدرية. وفي التقنيات الساكنة يقوم المخترقون بتفحص الشيفرة المصدرية للبرنامج بحثاً عن خطأ أو ثغرة أمنية ما، ففي البرمجيات مفتوحة المصدر يقومون بتفحص الشيفرة المصدرية المتاحة

لهذه البرمجيات، أما في البرمجيات المغلقة فإنهم يقومون بتفحص الشيفرة المغلقة (والتي تكون عادةً بلغة الآلة Machine Code)، وقد يستخدمون أيضاً أدوات تدعى "مفككات الترميز Decompilers" لإعادة الشيفرة الموجودة بلغة الآلة إلى الشيفرة المصدرية ومن ثمّ البحث ضمنها عن الثغرات الأمنية بنفس الأسلوب الذي سيتعاملون معه مع البرمجيات مفتوحة المصدر. هذه النقطة فائقة الأهمية: فعلى الرغم من أنّ المخترق قد يحتاج إلى استخدام الشيفرة المصدرية للبحث عن الثغرات الأمنية فإن البرمجيات المغلقة لا تتمتع بأية ميزة لأن المخترق قد يستخدم مفككات الترميز للحصول على الشيفرة المصدرية لهذه البرمجيات.

قد يتساءل البعض: ما زالت مفككات الترميز قادرة على إعادة شيفرة لغة الآلة إلى الشيفرة المصدرية الأصلية، لماذا يطلب المطورون على الدوام الشيفرة المصدرية للبرمجيات ولا يكتفون بالشيفرة المترجمة إلى لغة الآلة؟ المشكلة الأساسية هي أنه وعلى الرغم من عدم حاجة المطورين إلى الشيفرة المصدرية للبحث عن الثغرات الأمنية في برنامج ما فإنّهم سيحتاجونها لتطوير إمكانيات هذا البرنامج. ومع أنّ مفككات الترميز قادرة على على إعادة لغة الآلة إلى الشيفرة المصدرية الأساسية، فإنّ تعديل هذه الشيفرة المصدرية الناتجة أمر فائق الصعوبة، لأنه وأثناء عملية تفكيك الترميز فإن معظم التسميات المقروءة في الشيفرة المصدرية ستضيع، كما ستضيع جميع الملاحظات التوضيحية المضمنة في البرنامج. هذا الموضوع ليس ذا شأن عند البحث عن الثغرات الأمنية، لأن المخترق لا يهتم بتفاصيل الشيفرة المصدرية، وكل ما يبحث عنه هو ثغرة أمنية واحدة. أي أن مفككات الترميز تسهم في البحث عن أساليب لاختراق البرمجيات، ولمكنها ليست ذات فائدة في عملية تطوير هذه البرمجيات.

والخلاصة أن المطورين محقّون عند طلبهم الشيفرة المصدرية لتطوير البرمجيات، كما أنّ حجب الشيفرة المصدرية للبرمجيات المغلقة لا يسهم في حماية هذه البرمجيات من الإختراق.

لماذا لا يجدي إخفاء الثغرات الأمنية في التخلص منها؟

يعتبر البعض أنّ وجود ثغرة أمنية في برنامج ما دون معرفة ذلك سيمنع المخترقين من استغلال هذه الثغرة وبأنّ البرنامج سيبقى آمناً من الناحية العملية. نظرياً قد ينطوي هذا الكلام على شيء من الصحّة، ولكن المشكلة تكمن في أنّ هذه الثغرة وبمجرد اكتشافها من قبل أحد المخترقين ستستغل في محاولات هذا المخترق التخريبية عوضاً عن البحث عن حل لها. إنّ وجود ثغرات أمنية خفية لا يعني على الإطلاق أنّ هذه الثغرات ستبقى عصيةً على المخترقين، بل ستشكّل هذه الثغرات قنابل موقوتةً مزروعةً في البرنامج دون أدنى فكرة عن توقيت اكتشافها ولستغلالها. عملياً تعتبر مشكلة استغلال الثغرات الأمنية المكتشفة ضمن البرمجيات موجودةً في كل من البرمجيات المغلقة أو مفتوحة المصدر.

كما يؤمن البعض بأنّ أخبار الثغرات الأمنية يجب ألا تنشر وألا تناقش، وهو (نظرياً) تفكير سليم، ولكن في الواقع فإنّ المخترقين يقومون بنشر المعلومات عن الثغرات الأمنية بعدّة وسائل، أي أنّ هذا الأسلوب سيترك المطورين دون حماية دون أن يقدّم شيئاً في مواجهة المخترقين. لقد كانت الشركات عامةً في الماضي تخفي معلومات الثغرات الأمنية المكتشفة في برمجياتها، ولكن التجارب أثبتت أنّ الشركات بشكل عام لا تصلح الثغرات الأمنية في برمجياتها حتى يتعرف المستخدمون على هذه الثغرات ليقوموا بدورهم بالإصرار على الشركة لإصلاح هذه الثغرات.

كيف تواجه البرمجيات مفتوحة الأبواب الخلفية؟

من الشائع بأنّ البرمجيات مفتوحة المصدر، كونها غير مرتبطة بشركة واحدة تملك التحكم المطلق بها، عرضة لتضمين أدوات اختراق تدعى الأبواب الخلفية ضمن شيفرتها المصدريّة والتي ستمكّن المخترقين من الوصول إلى هذه البرمجيات بسهولة. صحيح أنّه من الممكن إدخال الأبواب الخلفيّة ضمن الشيفرة المصدريّة ولكنّها من الممكن أن تدخل ضمن شيفرة البرمجيات المغلقة أيضاً، قد يقوم أحد الموظفين المستائين لسبب ما أو لآخر بتضمين أبواب خلفية ضمن البرمجيات المغلقة، والتي سيكون احتمال إيجادها أقلّ بكثير من كون البرمجيات مفتوحة المصدر، فلا أحد سيستطيع تفحص الشيفرة المصدرية للبرنامج المغلق خارج الشركة المنتجة للبحث عن هذه الثغرات، وقلّة من الشركات تقوم بتفحص الشيفرة المصدرية لبرمجياتها بحثاً عن هذا النوع من الثغرات الأمنية. أما الإشاعة التي تفيد بأنّك تستطيع محاكمة الشركة المنتجة للبرنامج فهي مجرد أوهام، فجميع إتفاقيات الترخيص تحمي الشركات المنتجة للبرمجيات من المسؤولية القانونية.

لقد قامت شركة بورلاند Borland على سبيل المثال بتضمين باب خلفي عمداً في برنامجها الخاص بإدارة قواعد البيانات ”Interbase“ بين أعوام 1992 و 1994، يمكّن هذا الباب الخلفي أي مستخدم محلي أو بعيد بتعديل أيّ جزء من قاعدة البيانات وتشغيل برمجيات أخرى، كما قد يمنح المستخدم صلاحيات مدير النظام في بعض الأحيان0 بقيت هذه الثغرة الأمنية في البرنامج على مدى سنوات ستّة، لم يستطع أحد مراجعة الشيفرة المصدرية للبرنامج لاكتشافها، ولم يكن لدى بورلاند أي مبرر لحل هذه المشكلة. في حزيران من عام 2000 أطلقت بورلاند الشيفرة المصدرية لبرنامجها ليبدأ مشروع ”Firebird“ بالعمل مع

هذه الشيفرة المصدرية ليكتشف هذه الثغرة الأمنية الخطيرة في ''Interbase'' في كانون الأول من نفس العام. في كانون الثاني من عام 2001 أعلنت CERN وجود هذه الثغرة الأمنية، والمؤسف أنّه كان من الممكن اكتشاف هذه الثغرة بسهولة باستخدام إحدى أبسط التقنيات التي يستخدمها المخترقون. وبمجرّد اكتشاف هذه الثغرة من قبل المطورين الذين يعملون على تفحص الشيفرة المصدرية المفتوحة قاموا بحلّها بسرعة فائقة.

قد تعتبر أنّ إخفاء كلمة السر سيبقي نظامك آمناً، وبأنّ إتاحة الشيفرة المصدرية لبرنامج ما ستجعله أقلّ أمناً، ولكنّ ذلك غير صحيح، فالتقنية التي استخدمت لاكتشاف الثغرة الأمنية في برنامج بورلاند مستخدمة بشكل واسع في أوساط المخترقين، وبعض هؤلاء قادرون على إخفاء نجاحاتهم في اكتشاف الثغرات الأمنية بشكل ممتاز، ولا دليل لدينا أنّ هذه الثغرة لم تستغل لمرّات كثيرة. ولكن من الواضح أن الشيفرة المصدرية وبمجرد إتاحتها فقد تم تفحصها بدقّة مما أدّى إلى اكتشاف الثغرات الأمنية الموجودة ضمنها وحلّها.

عندما تتحول البرمجيات من مغلقة إلى مفتوحة المصدر فإنها ستكون أقلّ أمناً في البداية (بسبب الثغرات الأمنية الموجودة ضمنها)، ويمكن مع الوقت أن يتم اكتشاف وإصلاح الثغرات الأمنية في هذه البرمجيات لتصبح أكثر أمناً. أمّا البرمجيات التي تنشأ منذ البداية مفتوحة المصدر فإنّ طبيعة آليات تطوير المصادر المفتوحة كفيلة بضمان أمنها قبل أن تستخدم على نطاق واسع. ولكن لا بدّ أن ننتبه إلى أنّ مجرد تحويل البرنامج المغلق إلى برنامج مفتوح المصدر لا يكفي لجعل هذا البرنامج آمناً، كما أن مجرد كون البرنامج مفتوح المصدر لا يعني البتّة أنّه آمن.

- في البداية لا بدّ ن مراجعة الشيفرة المصدرية للبرنامج بحثاً عن الثغرات الأمنية المحتملة، والمشكلة هنا تكمن في التساؤل التالي: هل ستتم مراجعة الشيفرة المصدرية فعلاً؟ تلعب الكثير من العوامل دوراً في التقليل من شأن هذه المراجعة: فقد يكون البرنامج مستخدماً على نطاق ضيّق (وبالتالي لن يجد عدداً كافياً من المطورين لمراجعة شيفرته المصدرية) أو مكتوباً بلغة برمجيّة نادرة الإستخدام. من المؤكد مثلاً أن برنامجاً يقوم بتطويره شخص واحد لن يحصل على حقّه من المراجعة والتفحص. بشكل عام كلما استقطب البرنامج عدداً أكبر من المطورين كلما ازدادت إحتمالات إيجاد الثغرات الأمنية. وبما أنّ الحوارات والمعلومات المتعلقة بالبرمجيّات مفتوحة المصدر متاحة للجميع فإن المستخدم لأي برنامج مفتوح المصدر قادر على تقييم مستوى المراجعة والتنقيح لهذا البرنامج.

من أهمّ العوامل التي قد تعيق عمليّة المراجعة والتنقيح هي كون البرمجيّات ذات مصادر مغلقة. تدّعي بعض شركات تطوير البرمجيات أنّ برمجياتها مفتوحة المصدر، ولكنّ هذه البرمجيّات في الواقع (متاحة المصدر) أي أنّ المطور قادر على استعراض الشيفرة المصدريّة ولكنه لا يستطيع تعديلها، وبما أنّ مالك هذه البرمجيات يملك صلاحيات مطلقةً فإن المطورين لن يتشجعوا على العمل (مجّاناً) في تطوير هذه البرمجيات، حيث ستنخفض احتمالات المساهمة في عمل تطوعي عندما سيمتلك شخص آخر الحقّ في نتاج هذا العمل دون القائمين به، وكما يقول بروس بيرنز "من يريد أن يعمل لحساب شخص آخر دون مقابل؟".

- بعد ذلك لا بدّ أن يملك بعض المطورين مهارات تمكّنهم من بناء برمجيّات آمنة. فعدد الأشخاص الذين يقومون بمراجعة الشيفرة المصدرية بحثاً عن الثغرات الأمنية لا يهمّ ما لم يعرف بعضهم على الأقل كيفية التعامل مع هذه الثغرات عند

إيجادها. وليس من الضروري أن يمتلك كل من يقوم بمراجعة الشيفرة المصدريّة مهارات في أمن المعلومات مادام من يمتلك هذه المهارات متابعاً للعمليّة.

- وأخيراً لا بدّ عند العثور على الثغرات الأمنيّة أن تحل وأن يوزع هذا الحل ليكون متاحاً لجميع المستخدمين، وتمتاز البرمجيات مفتوحة المصدر بشكل خاص بسرعة حل المشكلات الأمنية التي قد تكتشف ضمنها. إنّ ضمان توفّر حلول جميع المشكلات الأمنية المكتشفة على أنظمة المستخدمين يعتبر مشكلةً لكل من البرمجيات المغلقة أو مفتوحة المصدر.

أسئلة عليك إجابتها

يكثر الحديث هذه الأيّام عن أمن المعلومات، ونسمع كل يوم أخباراً عن اختراقات لأنظمة الشركات والمؤسسات وسرقة معلوماتها، ومع هذا التزايد الواضح لأهميّة أمن المعلومات لا بدّ من العمل على تأمين أقصى درجة ممكنة من الحماية للبرمجيّات وأنظمة المعلومات، ولتقييم خياراتك فيما يتعلق باعتماد البرمجيّات مفتوحة المصدر ومدى أمن هذه البرمجيّات عليك أن تجيب على الأسئلة التالية:

- ما هي قيمة المعلومات التي ينبغي حمايتها؟ وما هو مدى أهميتها لعمل المؤسسة؟
- ما هي سمعة البرمجيّات التي ستستخدمها من الناحية الأمنيّة؟
- ما هو مستوى انتشار هذه البرمجيات وما هو حجم قاعدة المطورين الذين يعملون على تطويرها؟
- كم من الوقت بإمكانك أن تنتظر حتى تحل المشاكل الأمنيّة في البرمجيّات التي تستخدمها؟ وهل تلبّي هذه البرمجيات هذا المطلب؟

الفصل السابع

المصادر المفتوحة في قطّاع التعليم

مقدّمة

من أكثر الأمور غرابةً ومدعاةً للعجب أن نحاول إقناع القائمين على قطّاع التعليم بأن مشاركة المعرفة أفضل من حيث المحصلّة من إحتكارها، فقد تطورت الفنون والآداب والعلوم الإنسانية بنتيجة التشارك المنفتح والتفاعل للآراء والأفكار والأبحاث. ومع ذلك فإننا نجد الغالبية العظمى من الأنظمة المعلوماتية في مؤسساتنا التعليمية معتمدةً على البرمجيات المغلقة، وهو ما يضع القائمين على قطّاع التعليم في نفس خانة أصحاب شركات البرمجيات التجارية والذين يسهمون بشكلٍ ما أو بآخر في تقييد مدى المعرفة (على الأقل فيما يختص بالبرمجيات) التي يحصل عليها طلاب العلم.

ولا تتوقف مشكلات البرمجيات المغلقة على إخفاء آليّة عمل هذه البرمجيات، بل تتعدى ذلك إلى تقييد المستخدم (وهو هنا الطالب الذي يدرس هذه البرمجيات) عبر حصر المهارات التي سيحصل عليها أثناء دراسته بمنتجاتٍ برمجيةٍ تجاريةٍ سيصعب عليه الإستغناء عنها أو الإنتقال إلى غيرها على مدى حياته المهنية لأن ذلك سيتطلب منه الكثير من العناء والوقت. هذه البرمجيات عرضة للتغيير تبعاً لأهواء منتجيها والذين يدفعهم دوماً هاجس تحقيق المزيد من الأرباح.

بناء المستقبل

لقد حمّل القائمون على قطّاع التعليم عبر التاريخ مسؤولية الدفاع عن حق الوصول إلى المعلومات ومشاركتها ضد القوى المختلفة التي قد ترى مصلحتها في عكس ذلك. يبدو تأثير هذه القوى اليوم جلياً في تراخيص إستخدام البرمجيات المغلقة والتي تحدّ من قدرة المستخدم على التعلم والتطوير وتسهم في توسيع مساحة استغلال الشركات الإحتكارية. وكمثال حي على ما نحاول قوله هنا، لنتخيل مثلاً أنّ آلة غوتنبرغ (وهي أول آلة للطباعة في العالم) كانت مغلقة ولا يحقّ لأحد أن يطّلع على آلية أو مبدأ عملها، أين كنّا سنصل في تعليم مجتمعاتنا في ظلّ إحتكار آلة الطباعة؟

نحن نقف اليوم على أعتاب عصر المعلومات، والذي سيؤثر على العالم وعلى حياة مواطنيه بشكل يفوق بمرات عدّة تأثير عصر الثورة الصناعية. وينبغي اليوم أكثر من أي وقت مضى أن تفتح أدوات بناء هذا العصر وأن تتاح لجميع بني البشر ليسهم كل منهم بحصته في بناء هذا العصر. ومن المعيب أن نبني عقول أبنائنا في هذه المرحلة ليتحولوا إلى متلقين سلبيين للمعلومات، يعتمدون على الحلول المغلقة والمستوردة لمشاكلهم، بما يحول بينهم وبين حقّهم العادل في الإسهام ببناء مستقبلهم والذي هو جزء من مستقبل عصر المعلومات.

ينبغي أن نمنح طلابنا، في أضعف الأيمان، الفرصة ليتعرفوا على كيفية عمل الأدوات التي نضعها بين أيديهم، ليكونوا قادرين على بناء فهم أعمق لهذه الأدوات ليقوموا بالتالي بتطوير هذه الأدوات عندما يلهمهم خيالهم أو تتطلب ذلك حاجاتهم. ولا ينبغي علينا بأي حال من الأحوال أن نحدّ من آفاقهم عبر إغلاق أدوات بناء عصر المعلومات وتحذيرهم من استراق النظر إلى بواطن هذه الأدوات ومحاولة اكتشاف آليات عملها أو مشاركتها مع الآخرين.

تكامل المعرفة

لقد بني مجتمع العلوم منذ بداياته الأولى على أسس تشابه إلى حدٍ بعيد فلسفة المصادر المفتوحة، فعندما يبتكر أحد العلماء أو الباحثين شيئاً جديداً فإنه يحصل على التقدير والثناء على جهوده، ولكنه لا يمنح ملكية هذا الإبتكار، بل على العكس، يطلب من العلماء والباحثين أن يقوموا بنشر نتاجات أبحاثهم في منشورات مفتوحة متاحة للجميع. وعادة ما تقوم هذه المنشورات بنشر الأبحاث أو نتائجها بعد أن تتم مراجعتها من قبل عدد من الباحثين مهمتهم التأكّد من أصالة هذه الأبحاث ومطابقتها للمنهجيات العلمية المعتمدة.

بدأ الباحثون مؤخراً بإدراك أن اعتمادهم على البرمجيات المغلقة سيؤثر سلباً على منهجيتهم العلمية. فالباحثون يستخدمون البرمجيات في جوانب عدّة من نشاطاتهم العلمية، ولكنّهم نادراً ما يتيحون الوصول إلى هذه البرمجيات لغيرهم من الباحثين، وحتى في حال فعلوا ذلك، فإنّهم يستخدمون البرمجيات المغلقة التي تخفي شيفرتها المصدرية وتحمى بقوانين حماية الملكية الفكرية. هذا الأسلوب في العمل يتنافى وأبسط قواعد المنهجية العلمية، فلكي يتم التأكد من تكامل العمل المعرفي لا بد من أن يكون هذا العمل متاحاً للمراجعة والتحقق من قبل باحثين آخرين غير أولئك الذين قاموا بالعمل نفسه، وهو ما تحول دونه البرمجيات المغلقة.

ويذهب البعض إلى اعتبار العلماء والباحثين مجبرين على استخدام البرمجيات مفتوحة المصدر، وبأن مستقبل العلوم يعتمد بشكل كبير على قدرتهم في تحقيق ذلك. وبأن على الجامعات والمعاهد العلمية أن تتحرك بإتجاه إعتماد البرمجيات مفتوحة المصدر والتخلي عن البيئات البرمجية المغلقة المستخدمة حالياً.

مناهج تعليم تقنية المعلومات

يدّعي البعض أننا وسواء كنّا محبين لشركات إنتاج البرمجيات التجاريّة أم لا فإننا مجبرون على تعليم طلابنا كيفية التعامل مع هذه البرمجيات التي ستواجههم في مواقع عملهم في المستقبل. ومن المدافعين عن هذه النظرية الشركات التجارية المحليّة التي ترغب في التخلص من المبالغ الطائلة التي تنفقها على تدريب موظفيها الجدد على استخدام هذه البرمجيات بكفاءة. ولذلك فإننا نلاحظ أن مناهج تعليم تقنية المعلومات في المدارس والمعاهد والجامعات أشبه ما تكون دورات تدريبية معدّة من قبل شركات إنتاج البرمجيات التجارية. وبالتأكيد ستستغل الشركات التجارية هذه الفرصة لعقد الصفقات مع هذه المؤسسات التعليمية لتزويد طلابها بنسخ من البرمجيات التي تنتجها. هذه النظرية تجانب الحقيقة، فمن الخطأ أن نقوم بتدريس طلابنا البرمجيات المغلقة، عدا عن أن ذلك لا يلبّي أيضاً متطلبات الشركات التجارية للأسباب التالية:

- قد تحصل الشركات على وفر في نفقات التدريب إذا تم تدريس الموظفين في المدارس والمعاهد والجامعات كيفية التعامل مع البرمجيات التجارية المغلقة المستخدمة في العمل، ولكن هذا الوفر مؤقت، فمعظم شركات إنتاج البرمجيات التجارية تطرح بشكل دوري إصدارات جديدة من برمجياتها تحتوي على المزيد من المزايا والتعديلات، وبالتالي فإن الطلاب المتدربين على إصدار معين من هذه البرمجيات سيحتاجون إلى التدرّب مجددا على الإصدارات الجديدة.
- يعتمد بناء مجتمع المستقبل بشكل رئيسي على المواطن المثقّف، وهو ما يدعو المدارس والمعاهد والجامعات إلى العمل على بناء فهم أساسي واضح لتقنية المعلومات لدى الطالب يتجاوز تعليم نسخ الملفات وتعديلها ليتكون لدى الطالب

القدر اللازم من المعرفة لبناء آرائه الخاصة فيما يتعلق بقضايا تقنية المعلومات، بما فيها المخاطر الأمنية المترتبة على استخدام البرمجيات المغلقة على سبيل المثال.

- إنّ الإيقاع المذهل لتطور تقنية المعلومات بما فيها صناعة البرمجيات سيتسبب في فقدان جانب هام من المعلومات التي سيتلقاها الطالب أثناء دراسته بمجرد تخرجه! ينبغي على المدارس والمعاهد والجامعات بدلاً من أن تعلم طلابها مهارات استخدام برمجيات معينة أن تقوم بمنحهم المعرفة والفهم الأساسيين لتقنية المعلومات بما يمكنهم من تعلم كيفية تطبيق أية تقنية جديدة سيصادفونها بسرعة وسهولة.

يتضح لنا مما سبق أنّ مناهج تعليم تقنية المعلومات في المدارس والمعاهد والجامعات يجب أن تركز على تزويد الطلاب بفهم عميق لمتطلبات استثمار تقنية المعلومات بفاعلية وباستخدام أدوات عدّة. كما يجب أن تقدّم للطلاب فهماً معمّقاً لأساسيات مبادئ تقنية المعلومات.

ونظراً لأهمية مهارات تقنية المعلومات بالنسبة للطلاب، ولكل ما ذكرناه آنفاً، فإن على المؤسسات التعليمية ألا تبني مناهج تعليم تقنية المعلومات وفقاً لبرنامج معين، بل وفقاً للمبادئ الأساسية في استخدام التطبيقات البرمجية، فالطالب الذي يتمتع بفهم واضح لأساسيات الجداول البيانية قادر على استخدام مهاراته لاستثمار أي برنامج للجداول البيانية قد يصادفه. وعلى العكس، فالطالب الذي تعلم استخدام برنامج معين للجداول الحسابية لن يستطيع التعامل مع البرامج الأخرى أو حتّى الإصدارات الجديدة من نفس البرنامج دون أن يعاد تدريبه مجدداً على هذه البرمجيات أو الإصدارات الجديدة.

تلبّي البرمجيات مفتوحة المصدر جميع متطلبات المؤسسات التعليمية التي ستعتمد مناهج تعليم تقنية المعلومات لتدريس المبادئ الأساسية لتقنية المعلومات. حيث لا يوجد أي مبرر للتكاليف الباهظة المترافقة مع البرمجيات التجارية المغلقة لتحقيق هذه الغاية، هذا الوفر الكبير في تكاليف أنظمة تقنية المعلومات سيؤدّي إلى توسيع وتطوير إعتماد هذه الأنظمة على نطاق أكبر في المؤسسات التعليمية وإيصال مناهج تقنية المعلومات إلى عدد أكبر من الطلاب.

ردم الهوة الرقمية

إذا ابتعدنا قليلاً عن أجواء المؤسسات التعليمية ونظرنا إلى المجتمع بأكمله سنلاحظ أن الهوة بين المجتمعات التي تعتبر متقدمة تقنياً وتلك التي لا تملك المعرفة التقنية تزداد يوماً تلو الآخر، ولقد تضاعف الفرق بين مستوى دخل الفرد في أغنى دول العالم وأفقرها ثلاث مرات خلال الخمسة وثلاثين سنة الماضية. وبشكل عام ترتبط إمتلاك أدوات تقنية المعلومات والقدرة على التعامل مع هذه الأدوات هذه التقنية مع المرونة اللازمة للتفاعل مع المتغيرات الإقتصادية والإجتماعية، وتعود هذه المرونة بالكسب المادي على الدولة ال>ي ينعكس إيجابياً على مستوى دخل الفرد. وبالعكس، حيث يترافق الإفتقار إلى أدوات تقنية المعلومات وعدم القدرة على التعامل معها مع تناقص مستمر في مستوى دخل الفرد.

إذا كانت تقنية المعلومات تتحمل جزءا من المسؤولية عن هذه المشكلة، فهي قادرة دون شك على توفير الحلول لها، ومنها على سبيل المثال التعليم عن بعد بواسطة الإنترنت. يمكن للجامعات باستخدام تقنيات التعليم عن بعد أن تقوم بتوفير الفرص التعليمية للمواقع التي تعاني من نقص في الخدمات التعليمية في الوقت الراهن، كالمناطق الريفية والنائية. ولكن

التعليم عن بعد لن يؤدّي رسالته إذا وقع ضحية لرجال المال والأعمال والذين لا همّ لهم سوى تحقيق المزيد من الأرباح وهو ما سيدفعهم نحو اقتحام سوق التعليم عن بعد باعتباره سوقاً ضخماً جديداً.

إذا نظرنا بتمعن إلى الغاية الأساسية من التعليم عن بعد سنجد أن مصير هذه التقنية سيؤول إلى الفشل في حال تطّلب تطبيقها دفع مبالغ طائلة لقاء رسوم ترخيص استخدام البرمجيات التجارية والتجهيزات الحاسوبية اللازمة من قبل الطلاب والمؤسسات التعليمية، لأن التعليم عن بعد سيستثمر بشكل خاص لتوفير الفرص التعليمية للمناطق الريفية والنائية والتي تعاني أساساً من مشاكل مادّية.

يستحيل تبرير استخدام البرمجيات التجارية المغلقة في عملية التعليم عن بعد في ظل توفر بدائل مفتوحة المصدر تلبّي متطلبات هذه العملية. وهو ما دعى (على سبيل المثال) الحكومة المكسيكية لاختيار البرمجيات مفتوحة المصدر كأساس للبينة المعلوماتية في مؤسساتها التعليمية. ويعتبر استخدام البرمجيات التجارية المغلقة لتوفير خدمات التعليم عن بعد صفعةً توجّه إلى وجه الفقراء.

العدالة الإجتماعية

عندما تبدأ المؤسسات التعليمية في اعتماد البرمجيات مفتوحة المصدر فإنها ستلعب دوراً أساسياً في دعم استخدام هذه البرمجيات على مستوى العالم كوسيلة لبناء العدالة الإجتماعية. وستشكل محركاً أساسياً لدفع عجلة تطوير هذه البرمجيات مما سيؤدي إلى المساعدة في معالجة مشاكل التواصل مع تقنية المعلومات في جميع دول العالم.

تعاني المناطق النائية من العالم من مشاكل حقيقية تصطدم بها عند محاولة استخدام تقنية المعلومات، فعدا عن تكاليف تراخيص استخدام البرمجيات التجارية المغلقة والتي غالباً ما تكون هذه المناطق عاجزةً عن توفيرها فإن الشركات التجارية المصنعة لهذه البرمجيات لا تملك الرغبة في توفير الدعم لبرمجياتها في هذه المناطق. أما مجموعات تطوير البرمجيات مفتوحة المصدر فلا تقف نشاطاتهم عند تطوير برمجيات ذات نوعية عالية وحسب، بل يعملون على بناء شبكة من المطورين حول العالم لمشاركة الخبرات الضرورية لتطبيق هذه البرمجيات في ظل شتى الظروف والمناطق الجغرافية.

أي أن البرمجيات مفتوحة المصدر تقدّم المفتاح لمعالجة جانب كبير من مشاكل العدالة الإجتماعية في العالم، وذلك ينبغي على المدارس والجامعات والمعاهد أن تبذل ما بوسعها لتبنّي البرمجيات مفتوحة المصدر وتطويرها.

القرار الصعب

قد يشكّل الإنتقال إلى المصادر المفتوحة تحدّياً صعباً للمؤسسات التعليمية، حيث أن هذه البرمجيات تحتاج إلى بعض الوقت لتعلمها، كما أن تكاليف الدعم الفني ستزداد نتيجة عبث الطلاب بإعدادات البرمجيات الجديدة والتي ستحتاج إلى إعادة إعدادها بشكل صحيح. ولكن هذه الصورة تتغير بسرعة، فالتعامل مع البرمجيات مفتوحة المصدر يزداد سهولة يوماً تلو الآخر، والأهم من ذلك أن البرمجيات مفتوحة المصدر بطبيعتها تتيح للمؤسسات التعليمية تعديل هذه البرمجيات وبناء إصدارات خاصة بها، والتي قد تتضمن جميع البرمجيات اللازمة للعملية التعليمية.

ومن أهم العقبات التي ستواجه اعتماد المصادر المفتوحة في المؤسسات التعليمية المنافسة الشديدة التي تواجهها هذه البرمجيات من قبل الشركات المنتجة للبرمجيات التجارية المغلقة والتي تبذل ما بوسعها للهيمنة على سوق قطّاع التعليم، وقد تسرب في الآونة الأخيرة الكثير من المعلومات التي توضّح الإجراءات التي تقوم هذه الشركات باتخاذها لمواجهة المصدر المفتوحة وإبعادها عن المؤسسات التعليمية، تماماً كما تقوم هذه الشركات بمحاربة منافسيها من الشركات التجارية أيضاً.

تقوم بعض الشركات التجارية ببعض الإجراءات التي تساعدها على مواجهة البرمجيات مفتوحة المصدر مثل القيام بتعديل البروتوكولات المستخدمة حالياً لجعل التفاعل بين البرمجيات المغلقة والمفتوحة أكثر صعوبةً وكلفةً بهدف طرد البرمجيات مفتوحة المصدر من السوق. إن مجرد تفكير هذه الشركات باعتماد هذه الإجراءات يكفي (من وجهة نظري) لمنع هذه الشركات من الهيمنة على قطّاع التعليم، حيث يبدو أن هذه الشركات لا تدرك الفرق بين المنافسة الشريفة والإحتكار، والأسوأ من ذلك كله أن المستخدم وحده هو من يدفع تكاليف هذه الإجراءات الإحتكارية. وبما أن هذه الشركات تقوم بالعديد من الخطوات والإجراءات التي تتحدّى المبدأ الأساسي المعتمد في المؤسسات التعليمية لمشاركة الأفكار والمعلومات، فإن هذه المؤسسات ليست مضطرةً لخدمة مصالح الشركات التجارية، بل على العكس، ينبغي أن تتجه نحو اعتماد البرمجيات مفتوحة المصدر والتي تتوافق فلسفة تطويرها مع المبادئ الأساسية للبحث العلمي.

ما تقدّمه المصادر المفتوحة للمؤسسات التعليمية

تستطيع المؤسسات التعليمية الإفادة من برمجيات المصادر المفتوحة في مناح عدّة، منها على سبيل المثال لا الحصر:

تكاليف استخدام وتشغيل البرمجيات مفتوحة المصدر أقل من مقابلتها المغلقة

تعتبر تكاليف التشغيل الكلّية للبرمجيات مفتوحة المصدر أقل بكثير من تكاليف البرمجيات المغلقة، وقد بينت بعض الدراسات أن تكاليف استخدام نظام التشغيل مفتوح المصدر لينكس أقل بأربعين في المائة من تكاليف استخدام ويندوز، ويأتي الجزء الأكبر من هذا الوفر من تكاليف تراخيص استخدام هذه البرمجيات، وكلما كان عدد الحواسيب أكبر في المؤسسة كلما كان الوفر في التكاليف أكبر. كما يوفر استخدام البرمجيات مفتوحة المصدر في نفقات الصيانة والإدارة على الرغم من انخفاض تكاليف خبراء البرمجيات المغلقة، وذلك لتقلص الحاجة إلى عمليات الصيانة للبرمجيات مفتوحة المصدر.

التخلص من متاعب إدارة تراخيص إستخدام البرمجيات

لا بدّ لأي مؤسسة تعتمد البرمجيات التجارية المغلقة من أن تقوم وبشكل دوري من التحقق من مطابقة تراخيص استخدام البرمجيات التي حصلت عليها مع الواقع الفعلي لاستخدام هذه البرمجيات، وإلا فإنها ستعجز عن إثبات قانونية استخدامها لهذه البرمجيات. هذه المسألة قد تشكّل مشكلة حقيقية للمؤسسات التعليمية، حيث أن الشركات المنتجة للبرمجيات وهيئات الرقابة التي تدعمها تقومان بشكل مستمر بتفقد المؤسسات التعليمية للتحقق من قانونية

البرمجيات المستخدمة في هذه المؤسسات، وفي حال عدم توافق الرخص المدفوعة مع واقع استخدام هذه البرمجيات فإنها تفرض غرامات مالية ضخمة على هذه المؤسسات.

ونتيجة تكرار المشاكل القانونية الناتجة عن هذه المشكلة، ونظراً لصعوبة متابعة وإدارة تراخيص إستخدام البرمجيات في المؤسسات التعليمية، فقد حاولت الشركات التجارية تطوير صيغ خاصة لتراخيص استخدام البرمجيات المغلقة تتلاءم والمتطلبات الخاصة للمؤسسات التعليمية. في مثل هذه التراخيص تمنح الشركة المنتجة للبرنامج المغلق حقّ استخدام برنامجها لجميع الطلاب وأعضاء الهيئة التعليمية مقابل مبلغ معين تحصل عليه الشركة لكل طالب في هذه المؤسسة. من عيوب هذا الأسلوب في ترخيص استخدام البرمجيات المغلقة أنه يفرض على المؤسسة التعليمية دفع نفقات الترخيص كل سنة، مما يشكل عبئاً ثقيلاً على الموارد المالية للمؤسسة. كما أن حساب تكاليف الترخيص وفقاً لعدد الطلاب المسجلين في المؤسسة التعليمية غير عادل لأنه سيشمل طلاباً قد لا يستخدموا هذه البرمجيات على الإطلاق.

وفي المقابل تقدّم البرمجيات مفتوحة المصدر بديلاً أبسط وأكثر مرونةً: حيث يمكن للمؤسسة التعليمية أن تستخدم أي برنامج تريد لأي عدد من المستخدمين مجاناً!

البرمجيات مفتوحة المصدر أكثر أمناً ووثوقية

تتميز البرمجيات مفتوحة المصدر بمستوى رفيع من الأمن والوثوقية في مقابل البرمجيات المغلقة، لأن طبيعة تطوير هذه البرمجيات وإتاحة الوصول إلى الشيفرة المصدرية لها سيجعلها أكثر أمناً ومناعةً في وجه محاولات الإختراق والتخريب. ففي حين يحصل

مستخدموا البرمجيات المغلقة على "صندوق أسود" لا حول لهم في مواجهته سوى الأمل في أنّه لن يحتوي على ثغرات أمنية أو أبواب خلفية، فإن البرمجيات مفتوحة المصدر تتيح لمستخدمها تفحص شيفرتها المصدرية والتأكّد من مستوى الأمن الذي توفره، مما يعني أنّ أي مشكلة أمنية "في حال وجودها" ستكتشف وتتم معالجتها مباشرةً.

ولقد تعالت مؤخراً الأصوات المنادية باستخدام بدائل أكثر أمناً للعديد من البرمجيات المغلقة والتي تعرضت للعديد من الإختراقات، ومعظم هذه البدائل المقترحة كانت مفتوحة المصدر.

برمجيات المصادر المفتوحة ستساهم في القضاء على قرصنة البرمجيات

تشكّل قرصنة البرمجيات من قبل الطلاب مشكلةً حقيقيةً، وتظهر الدراسات أن الطلاب هم أكثر الفئات استخداماً للبرمجيات المقرصنة، وقد يكون أحد أسباب هذه المشكلة وجود بعض البرمجيات التي تطلبها المؤسسات التعليمية من طلابها دون تزويدهم بنسخ مرخصة منها لاستخدامها في منازلهم.

بإمكان المؤسسات التعليمية المساهمة في القضاء على هذه الظاهرة عبر تبني البرمجيات مفتوحة المصدر وتشجيع الطلاب على استخدامها كبديل للبرمجيات المغلقة، مما سيتيح لهم استخدام هذه البرمجيات متى شاءوا دون خرق قوانين حماية الملكية الفكرية.

البرمجيات المفتوحة المصدر كأداة تعليمية

ينطوي استخدام البرمجيات مفتوحة المصدر كأداة تعليمية على الكثير من الميزات في مقابل البرمجيات المغلقة، فعلى سبيل المثال يعتبر استخدام البرمجيات مفتوحة المصدر في تدريس مناهج تقنية المعلومات أكثر فاعليةً من استخدام البرمجيات المغلقة نظراً لتمكن الطلاب من الدخول إلى الشيفرة المصدرية لهذه البرمجيات ومعاينتها وتفهمها بشكل أفضل.

كما أن تدريب الطلاب على استخدام التطبيقات البرمجيات مفتوحة المصدر سيساعدهم على استيعاب مبادئ عمل هذه البرمجيات وليس فقط كيفية تشغيلها ليتمكنوا فيما بعد من التعامل مع أية تطبيقات برمجية تعتمد هذه المبادئ (بما فيها البرمجيات المغلقة).

حرّية الإختيار

لكي نكون أكثر واقعية علينا أن نعترف بأن إنتقال المؤسسات التعليمية إلى البرمجيات مفتوحة المصدر لن يتم بين ليلة وضحاها، ولا بدّ لضمان نجاح هذا الإنتقال أن نقوم بدراسة معمقة لواقع هذه المؤسسات وخصوصيات كل منها وأخذ هذه الخصوصيات بالحسبان أثناء التحول إلى المصادر المفتوحة. أي أننا سنواجه مرحلة تتواجد فيها البرمجيات مفتوحة المصدر جنباً إلى جنب مع البرمجيات التجارية المغلقة، ولا بدّ أيضاً من ضمان المواءمة والتفاعل بين هذه البرمجيات (وهو ما سنشرحه بالتفصيل في فصل آخر من هذا الكتاب) لمنح الطلاب وأعضاء الهيئة التدريسية الحرّية في اختيار البرمجيات التي يرغبون في التعامل معها وتعلمها.

لتحقيق هذه الغاية لا بدّ من تعديل بعض أساليب تعامل المؤسسة التعليمية مع تقنية المعلومات من حيث توفير التجهيزات والبرمجيات وتحديد معايير حفظ وتبادل البيانات والدعم الفني.

شراء التجهيزات

لا بدّ أن تلعب المؤسسات التعليمية (وبشكل خاص الجامعات والمعاهد العليا) دوراً فاعلاً في توفير التجهيزات الحاسوبية لطلابها وأعضاء هيئتها التدريسية، وقد تقوم هذه المؤسسات في سبيل ذلك بإبرام صفقات مع الشركات الموردة لهذه التجهيزات لتوفيرها للطلاب والمدرسين بأسعار مخفضة خاصّة مع تسهيلات بالدفع، ويسهم استخدام البرمجيات مفتوحة المصدر في هذا السياق بتخفيض أسعار هذه التجهيزات بشكل كبير نتيجة للتخلص من تكاليف تراخيص استخدام البرمجيات التجارية المغلقة والتي غالباً ما يدفعها المستهلك كجزء من سعر حاسوبه الجديد.

إن التنسيق بين المؤسسات التعليمية والشركات الموردة سيؤدّي إلى توفير حواسيب منخفضة الكلفة للطلاب والمدرسين، كما سيشكل ضماناً بأن البرمجيات مفتوحة المصدر المضمنة مع الحاسوب الجديد متوافقة مع المعايير التي ستضعها هذه المؤسسات وبالتالي سيحصل الطلاب والمدرسون على أعلى قيمة مقابل الأموال التي سيستثمرونها في شراء هذه التجهيزات.

معايير حفظ وتبادل البيانات

إذا حصل المستخدم على حرية الخيار بين استخدام البرمجيات مفتوحة المصدر أو البرمجيات المغلقة لابدّ أن تتجنب المؤسسات التعليمية اعتماد صيغ خاصة غير معيارية لحفظ أو تبادل البيانات بما قد يشكل تقييداً للحرية الممنوحة للمستخدم. وينبغي على هذه المؤسسات أن تلتزم باعتماد صيغ تبادل وحفظ البيانات المعيارية في جميع جوانب استثمارها لتقنية المعلومات، هذه الصيغ المعيارية ستضمن إمكانية المواءمة والتفاعل بين البرمجيات المفتوحة والمغلقة وتحمي حرية الإختيار.

الدعم الفنّي

وهو يشكّل العائق الأكبر الذي يحول دون تحول المؤسسات التعليمية (وغير التعليمية) إلى المصادر المفتوحة، عادةً يقوم المسؤولون عن الدعم الفنّي عندما تطلب منهم المساعدة في مشكلة تتعلق ببرنامج مفتوح المصدر بتوجيه صاحب المشكلة إلى مصادر الدعم الفنّي على الإنترنت ليبحث عن حل لمشكلته بنفسه.

إنّ التوجه نحو اعتماد البرمجيات مفتوحة المصدر يتطلب أن تقوم المؤسسات التعليمية بتوظيف أشخاص مهرة في توفير الدعم الفنّي لهذه البرمجيات وذلك لطمأنة المستخدمين بأن أية مشكلات قد تواجههم ستحل بسرعة ولن تتسبب في إيقافهم عن العمل.

ومن الضروري أيضاً أن تقوم المؤسسات التعليمية بتوفير قدر ملائم من التدريب للطلاب وأعضاء هيئة التدريسية على البرمجيات الجديدة لتجاوز الحاجز النفسي الناتج عن الخوف من المجهول لدى النفس البشرية.

كما يجب أن تتجنب المؤسسات التعليمية تضمين أية متطلبات خاصة في البرمجيات المستخدمة ضمن هذه المؤسسة، كأن تطلب مثلاً برنامجاً لتصفح الإنترنت لا يعمل إلا ضمن نظام تشغيل مغلق لاستخدامه في الوصول إلى موقع المؤسسة أو المواد التعليمية التي توفرها. ولا يعتبر تجاوز هذه المشكلة أمراً صعباً نظراً لتوفّر بدائل مفتوحة المصدر للغالبية العظمى من البرمجيات التجارية المغلقة.

أسئلة عليك إجابتها

إذا كنت مسؤولاً عن مؤسسة تعليمية، أو كنت مهتماً بقطّاع التعليم وأردت النظر في إمكانية التحول إلى البرمجيات مفتوحة المصدر لا بدّ أن تقوم بدراسة تفصيلية لميزات ومساوئ هذا التحول لكي يكزن قرارك أكثر موضوعية وواقعية، حاول أن تجيب على الأسئلة التالية:

- ما هو مستوى اعتماد مؤسستي على تقنية المعلومات في الوقت الراهن؟
- ما هي التأثيرات الإيجابية والسلبية لكل من البرمجيات مفتوحة المصدر أو المغلقة على منهجية البحث العلمي؟
- إلى أي مدى سيؤدّي اعتماد مناهج تعليم الصادر المفتوحة في تطوير مهارات الطلاب؟
- كيف يمكن استثمار البرمجيات مفتوحة المصدر لوفير الفرص التعليمية للمحرومين منها سواء لأسباب جغرافية أو إجتماعية؟
- ما هي الميزات الإقتصادية لإعتماد البرمجيات مفتوحة المصدر؟

الفصل الثامن

تجارب حكوميّة في المصادر المفتوحة

مقدّمة

في ظل المتغيرات المتسارعة في عالم أضحت فيه المعلومات أساس الإقتصاد والمجتمع ينبغي على المؤسسات الحكوميّة في القطّاع العام أن تتحول إلى الحكومة الرقمية لاستغلال ما تقدّمه من ميزات في خدمة المواطن والمجتمع. هذا التحوّل يتطلب الكثير من أنظمة تقنية المعلومات التي تبنى عليها هذه الحكومة إضافةً إلى أساليب العمل المتّبعة في المؤسسات الحكوميّة.

يشكّل هذا التحوّل من وجهة النظر الإقتصادية تحدّيات كبرى نظراً للحاجة إلى استثمارات ضخمة في تقنية المعلومات للمؤسسات الحكوميّة لكي ينجح هذا التحوّل. ولذلك لا بدّ من تقييم جميع الحلول المتاحة لاتخاذ الخيار الصحيح. سنحاول في هذا الفصل التعرّف إلى بعض التجارب الحكوميّة في المصادر المفتوحة لمحاولة الإجابة على السؤال التالي: إلى أيّ مدىً تستطيع البرمجيّات مفتوحة المصدر تلبية متطلبات الإنتقال الآمن إلى الحكومة الإلكترونيّة؟

متطلبات الحكومة الإلكترونيّة

إن الإعتماد الأساسي للحكومة الإلكترونيّة على تقنية المعلومات يوجب البحث بدقّة وتأن عن أفضل الحلول التي ستستخدم لبنائها. ما هي التكاليف المباشرة وغير المباشرة المترتبة على

كل من الخيارات المتاحة؟ كيف ستتعامل الحلول المعتمدة مع الأنظمة الأخرى؟ ما هو مدى إعتمادية الحلول المقترحة؟ كيف سيضمن النقل الآمن للبيانات بين الأنظمة المختلفة؟ ما هي التداعيات الإجتماعية المترتبة على كل من الخيارات المتاحة؟

أمن المعلومات

يعتبر أمن المعلومات من أهمّ مقومات الحكومة الإلكترونيّة، لأنّ ازدياد الإعتماد على التقنية سيعني زيادةً في كمّ المعلومات المخزّنة إلكترونيّاً، ولذلك لا بدّ من التأكد بأنّ البرمجيات المستخدمة لا تحتوي على أيّة ثغرات قد تتسبب في الإستيلاء على هذه المعلومات أو تخريبها.

العامل الإقتصادي

الغاية الأساسيّة للحكومة الإلكترونيّة هي رفع سويّة الخدمات المقدّمة للمجتمع، هذا التطوير في مستوى الخدمات المقدّمة عبر الإنترنت سيعود بالكثير من التوفير في القطّاع العام على المدى البعيد، فيما يلي بعض النقاط المتعلقة بالعامل الإقتصادي.

آلية العمل

لا يهدف التحوّل إلى الحكومة الإلكترونيّة إلى إضافة المزيد من الحلول المتقدّمة تقنياً إلى الإجراءات الحكوميّة المعتمدة حالياً، فالمزايا المترافقة مع اعتماد الحكومة الإلكترونيّة

تتوقف على إعادة هندسة آليات العمل في المؤسسات الحكوميّة لتقوم تقنيّة المعلومات بدعم وتبسيط آليات العمل الجديدة.

الحلول المتكاملة

لكي تكون الحكومة الإلكترونيّة فعّالةً لا بدّ أن تتمكن الأنظمة التي تعتمد عليها من التخاطب مع بعضها البعض، ومن هنا تبرز الحاجة إلى توحيد المعايير المستخدمة في هذه الأنظمة لإتاحة التكامل فيما بينها. كما سنرى لاحقاً فإنّ اعتماد المعايير المفتوحة أمر أساسي لنجاح هذا التحوّل. لا بدّ أيضاً من التركيز على كيفية حفظ وتبادل المعلومات الحكوميّة.

الإدارة

تقوم معظم المؤسسات الحكومية حالياً ببناء أنظمتها المعلوماتيّة، ومن أهمّ الأسئلة التي تطرح نفسها في هذا السياق ما يتعلّق بإدارة هذا التطوير: هل ينبغي أن تتم إدارته مركزيّاً (وهو ما قد يعني خسارة مرونة النظام في تلبية الإحتياجات المحليّة الخاصّة) أو أنه سيدار بطريقة لا مركزيّة (مما قد يعني أنّ نفس العمل سيتكرر مرّات ومرّات ويدفع ثمنه في كلّ مرّة).

الحكومة كمستهلك

عند شراء البرمجيات تنشأ علاقة وثيقة بين البائع والزبون، وغالباً ما تقود هذه العلاقة إلى الإعتماد على البائع بشكل كبير، فعندما يشتري الزبون برنامجاً ما فإنّه مجبر على الإعتماد

على البائع لتطوير أو تعديل هذا البرنامج. وقد تمت صياغة الكثير من الشروط لمواجهة الإعتماد المفرط على البائع ولكنّها ليست ذات فاعلية كبيرة. لا بدّ إذا أن تقوم الحكومة (والتي تمثل هنا دور الزبون) بدراسة كيفية التعامل مع الشركات التي ستبيعها البرمجيّات.

الحكومة كمستثمر

يشكّل حجم القطّاع العام جزءاً كبيراً من نشاطات الشركات التجاريّة، بما في ذلك صناعة تقنيّة المعلومات، وتحظّر معظم القوانين الحكوميّة تفضيل مورد على آخر، ونتيجةً للحجم الهائل للمشتريات الحكوميّة من تقنيّة المعلومات فإنّ البرمجيّات التي ستعتمدها الحكومة في حال تمّ تنظيم عمليّات الشراء بين مختلف المؤسسات الحكوميّة ستحصل على موقع ريادي في السوق، ولذلك لا بدّ من توجيه الإستثمارات اللازمة لعملية الإنتقال إلى الحكومة الإلكترونيّة في سبيل دعم صناعة تقنيّة المعلومات المحليّة.

ما هي أهميّة البرمجيات مفتوحة المصدر بالنسبة للحكومات

لقد اعتمد القطّاع الحكومي حتى وقتنا الراهن على البرمجيّات المغلقة بشكل أساسي، وتتعلق أهم الخيارات التي تواجه الحكومات حالياً بتوفّر برمجيّات مفتوحة المصدر تلبّي متطلبات العمل في المؤسسات الحكوميّة ومدى سهولة استخدام هذه البرمجيات بالمقارنة مع البرمجيات المغلقة وفيما إذا كانت هذه البرمجيات ستقدّم حلولاً أكثر فاعليةً لاحتياجات الإنتقال إلى الحكومة الإلكترونيّة.

تحدّثنا في الفصول السابقة عن أهميّة البرمجيّات مفتوحة المصدر بشكل عام، ولكننا سنوجز فيما يلي أهمّ ميزات هذه البرمجيّات فيما يتعلّق بخصوصية المتطلبات الحكوميّة:

- التوفير في التكاليف: البرمجيات مفتوحة المصدر ليست بالضرورة مجانيّة، ولكنّها في معظم الحالات أقل كلفةً من البرمجيّات المغلقة، وهذا يعني وفراً كبيراً من وجهة النظر الإقتصاديّة عند استخدام هذه البرمجيّات في المؤسسات الحكوميّة.
- المعايير المفتوحة وتكامل الأنظمة: من أهم خصائص البرمجيّات مفتوحة المصدر أنّها مبنيّة باستخدام معايير ومقاييس مفتوحة، أي أنّ مبادئ تطوير هذه البرمجيّات متاحة للجميع، بعكس البرمجيّات المغلقة والتي تستخدم معايير خاصّةً بكل شركة على حدة (وغالباً ما تحجب الشركة هذه المعايير عن الآخرين باعتبارها سراً فائق الأهميّة). ينبغي أن تتيح الحكومة الإلكترونيّة تبادل المعلومات والتفاعل معها بين الأنظمة المختلفة بسهولة ودون عوائق، وهو ما سيضمنه الإعتماد على المعايير المفتوحة.
- حماية المعلومات: تتطلّب الحكومة الإلكترونيّة تخزين المعلومات الحكوميّة بصيغة إلكترونيّة، مما يستوجب الإعتماد على أنظمة تقاوم قدر الإمكان محاولات التخريب أو سرقة المعلومات. إنّ توفّر الشيفرة المصدرية مع البرمجيّات مفتوحة المصدر يتيح تصفح هذه الشيفرة بحثاً عن أيّة ثغرات أمنيّة وحلّها مباشرةً. كما يستحيل في حالة البرمجيّات مفتوحة المصدر إخفاء أبواب خلفية عمداً أو بطريق الخطأ.
- التحرر من تحكّم الشركات التجارية: عندما تحصل إحدى المؤسسات الحكوميّة على برنامج ما فإن الشركة المنتجة لهذا البرنامج ستصبح قادرةً بطريقة ما أو بأخرى على التحكّم بهذه المؤسسة، وخصوصاً فيما يتعلّق بالإصدارات الجديدة أو التعديلات لهذا البرنامج. أمّا البرمجيّات مفتوحة المصدر فإنّها تحرر المؤسسات

من هذه الهيمنة كونها مبنيّة باستخدام معايير مفتوحة ولا تستخدم أية صيغ مغلقة لحفظ المعلومات، مما يجعل المستخدم قادراً على اختيار البرنامج والمورد الذي يريد.

- إمكانية التعديل والتطوير: تستطيع المؤسسات الحكوميّة بفضل توفر الشيفرة المصدرية للبرمجيّات القيام بأيّة تعديلات قد تراها ضروريةً لتحسين أداء هذه البرمجيات ولتلبية الإحتياجات الخاصّة بكل مؤسسة. كما تستطيع إعادة استخدام هذه النسخ المطورة من البرمجيّات في مؤسسات أخرى مما سينعكس إيجابياً على التوفير في تكاليف تطوير البرمجيّات.

المصادر المفتوحة من وجهة نظر حكوميّة

لقد خطفت البرمجيات مفتوحة المصدر الأضواء في عالم تقنيّة المعلومات مؤخراً، مما لفت انتباه العديد من الحكومات إلى ضرورة تقييم هذه البرمجيّات وتحديد مدى صلاحيتها للإستخدام في القطّاع الحكومي وما قد تقدّمه من ميزات وفوائد، وخلصت إلى اعتبار التحول إلى البرمجيّات مفتوحة المصدر ضرورةً ملحّةً لا بدّ من العمل على تحقيقها. وقد قامت بعض الدول فعلياً بإطلاق مبادرات لاستخدام البرمجيّات مفتوحة المصدر في مؤسساتها الحكوميّة.

ألمانيا

عقدت الإدارة المركزية الألمانيّة في شهر حزيران من عام 2002 إتفاقاً مع كلّ من IBM و SuSE لتزويدها ببرمجيّات مفتوحة المصدر تعتمد على نظام التشغيل لينكس، يتيح هذا

الإتفاق للإدارة الحصول على أنظمة تعتمد البرمجيّات مفتوحة المصدر بأسعار مخفّضة من IBM، بما فيها المخدّمات ومحطّات العمل، وتقوم IBM في المقابل بتوفير الدعم الفنّي لهذه الأنظمة. وترغب الحكومة الألمانيّة بتشجيع استخدام الحلول البديلة لأنظمة مايكروسوفت Microsoft مع هذا الإتفاق، ولكنّها لن تستصدر قانوناً بهذا الخصوص، بل ستعتبره خياراً متاحاً لصنّاع القرار.

المملكة المتحدة

أصدرت الحكومة البريطانيّة سياسيةً عامّةً في مجال المصادر المفتوحة في تموز 2002، وتنص هذه السياسة على أنّ الحكومة والسلطات البريطانيّة ستتعامل مع البرمجيّات مفتوحة المصدر على قدم المساواة مع البرمجيّات المغلقة عند شراء أنظمة تقنيّة المعلومات. كما تهدف هذه السياسة إلى استخدام المعايير المفتوحة قدر الإمكان في مؤسسات الحكومة البريطانيّة. من حيث المبدأ فإنّ الحكومة البريطانيّة ترغب بالحصول على أقصى ما يمكن الحصول عليه مقابل المبالغ التي تستثمرها في قطّاع تقنيّة المعلومات، وهو أحدّ أهم أسباب الإعتماد على البرمجيّات مفتوحة المصدر. كما ترغب الحكومة البريطانيّة بالتحرر من سيطرة الشركات التجاريّة. سنبحث في فقرة لاحقة التجربة البريطانيّة في المصادر المفتوحة بشيء من التفصيل.

فرنسا

إتّخذت الحكومة الفرنسية قراراً يقضي بإنهاء تعاقد الإدارة المركزيّة الفرنسيّة مع شركة مايكروسوفت لتوريد واستخدام منتجاتها، مما يعني أنّ على جميع السلطات المحليّة والوطنيّة الفرنسيّة أن تستخدم البرمجيّات مفتوحة المصدر قدر المستطاع.

شكّلت الحكومة الفرنسيّة مكتباً خاصّاً أطلقت عليه اسم "وكالة تقنيّة المعلومات والإتصالات في المؤسسات الحكوميّة" لمتابعة تنفيذ هذا القرار إضافةً إلى تنسيق جهود ومبادرات المؤسسات الحكوميّة في مجال تقنيّة المعلومات. على الوكالة أن تتأكّد من إعتماد المعايير المفتوحة في جميع المؤسسات الحكوميّة الفرنسيّة وأن تعمل على ضمان التوافقيّة بين أنظمة تقنيّة المعلومات وتخفيض تكاليفها.

وترغب الحكومة الفرنسيّة أيضاً في إتاحة الفرص للشركات الصغيرة العاملة في تطوير البرمجيّات عبر دعمهم للعمل في المشاريع الحكوميّة لتطوير البرمجيّات مفتوحة المصدر.

دول أخرى

أطلقت عدّة دول مبادرات تتعلق باعتماد البرمجيّات مفتوحة المصدر في عدّة مناطق من العالم. أصدر البرلمان الفنلندي على سبيل المثال توصيات باعتماد البرمجيّات مفتوحة المصدر في المؤسسات الحكوميّة. وقد شهدت البيرو الكثير من النقاشات حول استصدار قوانين تفرض على المؤسسات الحكوميّة استخدام البرمجيّات مفتوحة المصدر حصراً. أمّا جمهوريّة الصين الشعبية فقد اعتمدت منذ سنوات سياسةً وطنيةً لاعتماد البرمجيّات مفتوحة المصدر في جميع مؤسساتها الحكوميّة.

الحكومة البريطانيّة والمصادر المفتوحة

ذكرنا في الفقرة السابقة أنّ الحكومة البريطانيّة أصدرت سياسةً عامّةً لاعتماد البرمجيّات مفتوحة المصدر، وفيما يلي أهمّ النقاط الواردة في هذه السياسة:

- ستقوم الحكومة البريطانيّة بمعاملة البرمجيات مفتوحة المصدر على قدم المساواة مع البرمجيات المغلقة عند شراء كل ما يتعلّق بتقنيّة المعلومات، وسيتم اختيار الحل الأرخص ثمناً.
- ستقوم الحكومة البريطانيّة باختيار المنتجات التي تعتمد المعايير والمقاييس المفتوحة حصراً في جميع التطويرات المستقبليّة.
- ستتجنّب الحكومة البريطانيّة قدر الإمكان الإرتباط بمورد وحيد للمنتجات والخدمات.
- ستحاول الحكومة البريطانيّة الحصول على جميع الحقوق المتعلقة بالشيفرة المصدريّة للبرمجيّات التي ستقوم بشرائها.
- ستبحث الحكومة البريطانيّة في إمكانيات إعتماد البرمجيات مفتوحة المصدر كخطّة استثمارية في جميع البرمجيّات المطورة بتمويل من الحكومة.

أما مبررات إعتماد هذه السياسة فسنورد فيما يلي محاورها الأساسيّة:

- يجب أن يتم البحث دوماً عن الحلول التي تقدّم أكبر قيمة مقابل المبالغ المدفوعة ثمناً لها، هذه الحلول قد تكون مفتوحة المصدر أو مغلقة أو حتى خليطاً من كليهما، يجب أن تيم اتخاذ القرارات المتعلقة بكلّ حالة على حدة.

- لا بدّ من تلبية متطلبات التوافقية والتفاعل بين الأنظمة المختلفة لضمان نجاح مبادرة الحكومة الإلكترونيّة.
- يجب العمل على تخفيض تكاليف الأنظمة المستخدمة في المؤسسات الحكوميّة، ويساعد استخدام هذه السياسة في تحقيق هذا الهدف عبر:
  - الحصول على أكبر قيمة مقابل المبالغ المدفوعة ثمناً للأنظمة المعلوماتيّة.
  - التحرر من تبعات الإعتماديّة على مورّد وحيد.
  - توفير المزيد من المرونة في تطوير وتكامل وتفاعل الأنظمة المعلوماتيّة.
- يعتبر أمن البرمجيّات المستخدمة في المؤسسات الحكوميّة أمراً فائق الحساسيّة، وتتمتع البرمجيّات مفتوحة المصدر بسمعة مميزة في هذا المجال، وفي بعض الأحيان قد تكون البرمجيّات المغلقة غير ملائمة على الإطلاق للإستخدامات الحكوميّة نتيجة إفتقارها إلى القدر اللازم من الأمن.

تجربة البيرو مع المصادر المفتوحة

ينص دستور البيرو في الجزء الخامس من الفقرة 2DEG منه على أنّ "يملك جميع المواطنون الحقّ في الحصول على المعلومات التي يحتاجونها دون ذكر الأسباب، ويجب أن تقوم الجهات الحكوميّة المعنيّة بإيصال هذه المعلومات إلى طالبها بكلفة معقولة وضمن الفترة الزمنيّة التي يحدّدها القانون. يستثنى من هذه المادّة المعلومات التي قد تؤثر على خصوصيّة المواطنين أو الأمن القومي والتي سيحدّدها القانون."

كما يؤكّد الجزء السادس من نفس المادّة على حقّ جميع المواطنين "في ضمان أنّ جميع خدمات المعلومات سواء كانت مؤتمتةً أم لا في القطّاعين الخاص والحكومي لا تقوم بنشر المعلومات التي قد تؤذي الخصوصيّة الشخصية أو العائلية".

أي أنّ دستور البيرو مهتم بحماية حريّة المواطن في الحصول على المعلومات التي يحتاجها وفي نفس الوقت حماية المعلومات الشخصيّة الخاصّة به أو تلك التي قد تؤثّر على الأمن القومي.

ولا يمكن الإعتماد على حسن نيّة القائمين على تنفيذ الدستور وحسب لضمان تطبيق الآليات التي ستوفّر الحماية لحقوق المواطن، بل لا بدّ من الإعتماد على تقنيّات تسهم بشكل فعّال في حماية الحقوق المذكورة.

ويأتي قرار إعتماد البيرو على استخدام المصادر المفتوحة في جميع مؤسساتها الحكوميّة في سياق المساهمة في حماية حقوق المواطنين مبنياً على مبادئ أساسيّة يمكن تلخيصها فيما يلي:

- حريّة حصول المواطن على المعلومات العامّة.
- صحّة المعلومات العامّة.
- أمن المواطن والدولة.

لحماية حقّ المواطن في حريّة الحصول على المعلومات لا بدّ من التخلّص من السيطرة الإحتكارية للشركات التجارية على أسليب تخزين هذه المعلومات، ويمكن ضمان ذلك عبر

اعتماد صيغ حفظ المعلومات المعيارية والمفتوحة والتي تتيح أيضاً ضمان مستويات أعلى من التوافقية بين البرمجيّات المختلفة.

ولا يكفي في سبيل ضمان صحّة المعلومات العامّة أن يعتمد على الشركات التجاريّة لصيانة وتطوير البرمجيّات، خصوصاً في الحالات الإحتكاريّة.

ولضمان الأمن القومي لا بدّ من التحقق بأنّ الأنظمة المستخدمة في المؤسسات الحكوميّة خالية تماماً من أيّة ثغرات أمنيّة قد تتيح التحكّم بها عن بعد أو تسريب معلومات سريّة لأطراف لا ينبغي بهم الحصول عليها. ولذلك لا بدّ من اعتماد أنظمة يمكن الوصول إلى شيفرتها المصدريّة بما يوفّر إمكانية تفحصها والتحقق من أمنها.

تريد حكومة البيرو في سياق العمل على تطوير أداء إداراتها ومؤسساتها باعتبارها مصدر حفظ وإدارة المعلومات العامّة أن تحدّد الشروط التي ستنظم عمليات شراء البرمجيّات في جميع مؤسساتها الحكوميّة في المستقبل لتلبيّة متطلبات حماية حريّات المواطنين المنصوص عنها في الدستور والمبادئ الأساسية المذكورة آنفاً.

تؤكّد مبادرة حكومة البيرو على أنّه لا يكفي اعتماد أي برنامج سيستخدم في المؤسسات الحكومية بمجرّد تلبيته للمتطلبات الفنيّة لهذه المؤسسات، بل يجب أن تتحقّق بعض الشروط في إتفاقيّة الترخيص المرافقة لهذا البرنامج والتي لن تستطيع الحكومة دونها ضمان قدرة المواطن على معالجة المعلومات بشكل صحيح أو التحقّق من أمنها ووثوقيتها. تضمن هذه المبادرة أيضاً المساواة التامّة في التعامل مع جميع المواطنين، لأنّ البرمجيات مفتوحة المصدر متاحة للجميع دون استثناء.

سنتحدّث فيما يلي عن بعض الميزات التي ستحصل عليها الحكومة من جرّاء تنفيذ هذه المبادرة:

بادئاً ذي بدأ، ستوفّر هذه المبادة الكثير من فرص العمل للمبرمجين المحليين، فقد احتلّت البرمجيات مفتوحة المصدر 2% من مبيعات برمجيات المخدّمات في الولايات المتّحدة في عام 2001، وهو رقم كبير جداً بالنظر إلى السوق الأمريكية وحجم التنافس فيها، ويبدد تخوفات البعض من أنّ البرمجيّات مفتوحة المصدر تشكل تهديداً لسوق عمل المبرمجين، بل على العكس من ذلك فإنّ هذه المبادرة ستشجّع على تحرير كميّات هائلة من الموارد التي ستستثمر في إطلاق العنان للمطورين المبدعين. إنّ استخدام البرمجيّات مفتوحة المصدر سيمكّن محترفي تقنيّة المعلومات من معالجة مشاكل هذه البرمجيّات بشكل أفضل ورفع مستوى تطويرها مستخدمين لذلك الموارد الضخمة المتوفرّة في عالم المصادر المفتوحة.

من جهة أخرى فإنّ اعتماد هذه المبادرة سيقلّص من الإستخدام غير القانوني للبرمجيّات المقرصنة في بعض المؤسسات الحكوميّة، إنّ هذا الإستخدام غير القانوني أو حتى مجرّد التفكير بذلك يعتبر سبباً وجيهاً لحل هذه المشكلة التي تتعارض مع حقوق حماية الملكيّة الفكريّة والتي ستتسبب عاجلاً أم آجلاً في أذية إقتصاد البيرو. كما يمكن استخدام هذه المبادرة للتخلّص من الإستخدام غير القانوني للبرمجيّات المقرصنة في شركات القطّاع الخاص.

على الرغم من أنّ الإعتماد على البرمجيّات مفتوحة المصدر سيعود على إقتصاد البيرو بالوفر الكبير إلا أنّ العامل الإقتصادي لا يشكّل المحور الأساسي للمبادرة، بل تركّز هذه

المبادرة على أهميّة وجود ضمانات تقنيّة تتيح التعامل مع المعلومات العامّة بطريقة تتلاءم ومتطلبات الدستور فيما يتعلق بحماية حقوق المواطن.

يمكن تلخيص أهم الفوائد الناتجة عن تطبيق هذه المبادرة في النقاط التاليّة:

- الأمن القومي:

ينبغي أن تقوم الحكومة بحفظ المعلومات المتعلّقة بالمواطنين لكي تؤدي مهامّها بالشكل الأمثل، وترتبط العلاقة بين المواطن والحكومة ارتباطاً وثيقاً بأمن ودقّة هذه المعلومات التي يجب حمايتها على الدوام من الأخطار التاليّة:

  - تسريب المعلومات: يجب الحفاظ على المعلومات السريّة بطريقة تتيح الوصول إليها من قبل الأشخاص والمؤسسات المخوّلة بذلك فقط.
  - عدم التمكّن من الحصول على المعلومات: يجب أن تحفظ المعلومات بطريقة تضمن وصول الأشخاص أو المؤسسات المخوّلة بذلك إلى هذه المعلومات طيلة فترة صلاحيتها.
  - تحريف المعلومات: يجب حماية المعلومات في مواجهة تحريفها أو تعديلها إلا من قبل الأشخاص أو المؤسسات المخوّلة بذلك حصراً.

إن اعتماد البرمجيّات مفتوحة المصدر يتيح للمستخدم تفحص آليّات عمل هذه البرمجيّات، وهو ما يعتبر عاملاً حاسماً في أمن المعلومات لأنّه يقلل من إحتمالات احتواء هذه البرمجيّات على ثغرات أمنيّة أو أبواب خلفية.

- الإستقلالية التقنيّة:

تحدّ البرمجيّات التجاريّة المغلقة في غالبيتها من حريّة تطويرها أو تعديلها من قبل المستخدم، مما يعني ضرورة اعتماد مستخدم هذه البرمجيّات على الشركات المنتجة لها، وهو ما يضع هذه الشركات في موقع تستطيع بفضله إملاء شروطها

ومتطلباتها.

في المقابل، تمنح البرمجيات مفتوحة المصدر المستخدم مطلق الحريّة في تطويرها أو تعديلها كيفما يشاء لتلبية احتياجاته الخاصّة. ولا تقتصر الإفادة من هذه الحريّة على المبرمجين وحسب بل تمتد إلى المستخدم العادي الذي سيصبح بمقدوره الإعتماد على أيّ مبرمج (وليس بالضرورة المطور الأساسي للبرنامج) لتطوير هذا البرنامج أو إصلاح الخلل فيه.

- التطوير محلياً:

تمنح البرمجيّات التجاريّة المغلقة المستخدم الحريّة في تشغيل هذه البرمجيات دون أن يستطيع التعرّف عليها بعمق أو تعديلها، وهو ما يجعل المستخدم أسير تقنيّات لا يعرف آلية عملها. وبما أنّ آلية عمل البرنامج تعتبر سرّاً يجب إخفاؤه بشتّى الوسائل فإنّ إصلاح الخلل في هذه البرمجيّات عند ظهوره أمر شبه مستحيل، وهو ما يضع محترفي تقنيّة المعلومات المحليين في موقع يجدون أنفسهم فيه عاجزين عن تقديم أيّة قيمة مضافة مما يؤدّي بالتالي إلى حسر آفاق تعلمهم وتطورهم المهني. وتغيب هذه المشاكل كليّةً في عالم البرمجيّات مفتوحة المصدر.

- كلفة البرمجيّات:

تلغي البرمجيّات مفتوحة المصدر الحاجة إلى دفع رسم تراخيص الإستخدام والتي تشكّل عبئاً حقيقياً على الإقتصاد الوطني، خصوصاً مع الإنتشار المتزايد للإنترنت ومتطلبات التحوّل إلى الإقتصاد الرقمي. ينبغي أن تكون المؤسسات الحكوميّة قادرةً على التحكّم بنفقاتها في مجال تقنيّة المعلومات لكي لا تقع في مطبّات مستقبليّة تحول دون تمكنّها من تحقيق أهدافها.

إنّ تكاليف الإنتقال إلى البرمجيّات مفتوحة المصدر محدودة بتكاليف عملية الإنتقال ذاتها، والتي قد تتضمن تكاليف التدريب والتشغيل وتحويل البيانات والتطوير والوقت، هذه التكاليف غير ثابتة وليست مجرّد مبالغ تدفع مرّة واحدة وحسب. وفي المقابل يترافق استخدام البرمجيّات الجاريّة المغلقة مع تكاليف إضافيّة أيضاً عدا عن تكاليف إتفاقيات ترخيص الإستخدام، كتكاليف الحصول على الإصدارات الجديدة (والتي غالباً ما تتحكّم بإيقاعاتها الشركات الإحتكاريّة)، عدا عن الثمن الباهظ التي ستتكبده الحكومة نتيجة خسارة الحريّات التي تضمن تحكّمها في المعلومات التي تملكها. هذه التكاليف دائمة، ومع مرور الوقت سيتجاوز حجمها حجم تكاليف التحوّل إلى البرمجيّات مفتوحة المصدر. وباختصار شديد فإنّ الفوائد المجنيّة من التحوّل إلى البرمجيّات مفتوحة المصدر تتجاوز بمراحل تكاليف هذا التحوّل.

خلصت حكومة البيرو إلى اعتبار البرمجيّات مفتوحة المصدر أكثر الحلول التقنيّة ملاءمةً لحماية حريّات المواطنة المنصوص عنها في الدستور وقررت اعتماد هذه البرمجيات في جميع مؤسساتها.

ونظراً لأهميّة تجربة البيرو مع البرمجيّات مفتوحة المصدر نتيجة لتقارب ظروف هذا البلد من ظروف معظم الدول العربيّة فإننا سنورد فيما يلي النص الكامل لقانون إعتماد البرمجيات مفتوحة المصدر الذي أقرّه مجلس الشعب في البيرو.

نص قانون استخدام البرمجيّات الحرّة في البيرو

مجلس الشعب في جمهوريّة البيرو يقرّ ما يلي:

قانون استخدام البرمجيات الحرّة في المؤسسات الحكوميّة

الفقرة 1– الهدف من القانون:

تقييد استخدام جميع المؤسسات الحكوميّة في أنظمتها وتجهيزاتها الحاسوبيّة بالبرمجيّات الحرّة حصراً.

الفقرة 2– مواقع تطبيق القانون:

جميع الهيئات الإداريّة والقانونيّة والقضائيّة إضافةً إلى المؤسسات المستقلّة المحليّة والإقليميّة غير المركزيّة والشركات الأخرى التي تملك الحكومة الحصة الأكبر منها ستعتمد البرمجيّات الحرّة في أنظمتها وتجهيزاتها الحاسوبيّة.

الفقرة 3– الجهة المسؤولة عن تطبيق القانون:

يعتبر مجلس الوزراء السلطة المسؤولة عن تنفيذ هذا القانون.

الفقرة 4– تعريف البرمجيّات الحرّة:

تعرّف البرمجيّات الحرّة والتي يشملها هذا القانون بأنّها البرمجيّات التي تضمن للمستخدم دون أيّة تكاليف إضافيّة ما يلي:

- حريّة استخدام البرنامج لأي غرض.
- حريّة الحصول على الشيفرة المصدرية للبرنامج.
- حريّة تفحص آلية عمل البرنامج.
- حريّة استخدام آليات عمل البرنامج أو أيّة أجزاء منه لتطويعها في تلبيّة متطلبات المستخدم.
- حريّة نسخ وإعادة توزيع البرنامج.
- حريّة تعديل البرنامج وحريّة توزيع هذه التعديلات ضمن نفس إتفاقيّة ترخيص البرنامج الأصلي.

الفقرة 5- الإستثناءات:

عند عدم توفّر برنامج حرّ يقوم بتلبيّة الحاجة يمكن للمؤسسات الحكوميّة استخدام البدائل التالية وفقاً لتسلسل ورودها.
عند وجود قيود زمنية لحل مشكلة تقنيّة ما وكان هذا الحل متوفرّاً في برنامج مغلق ينبغي أن تقوم الجهة التي تحتاجه بطلب استثناء من الجهة المسؤولة عن تنفيذ هذا القانون لاستخدام البرنامج المغلق الذي يجب أن تتوفّر فيه الشروط التالية:

- يجب أن يتوافق البرنامج مع الشروط المذكورة آنفاً في الفقرة 4 من هذا القانون باستثناء حريّة إعادة توزيع البرنامج بعد تعديله. في مثل هذه الحالة يجب أن يكون الإستثناء محدداً.
- إذا لم تتوفّر أيّة برمجيّات من النوع السابق يجب عندها اختيار البرمجيات المتوفّرة ضمن مشروع متطوّر للبرمجيّات الحرّة. سيكون الإستثناء في هذه الحالة مرحليّاً ينتهي تلقائيّاً عند وصول البرنامج الحر البديل إلى تلبيّة الحاجة المطلوبة.

- إذا لم توجد أيّة برمجيّات تتحقق فيها هذه الشروط يمكن عندها استخدام البرنامج المغلق ولكنّ الإستثناء الذي ستصدره الجهة المسؤولة عن تطبيق هذا القانون سينتهي تلقائياً بعد سنتين من تاريخ إصداره، ويتطلّب تجديد الإستثناء إثبات عدم وجود أيّ بديل ملائم مفتوح المصدر.

ستقوم الجهة المسؤولة عن تطبيق هذا القانون بإصدار تصريح بالإستثناء فقط في حال تعهد الجهة الحكوميّة بحفظ البيانات بصيغ قياسيّة، ودون التأثّر بدفع تكاليف تراخيص الإستخدام.

الفقرة 6- التصاريح التعليميّة

يجب أن يسمح لكل مؤسسة تعليميّة حكوميّة بالتقدم بطلب استثناء لاستخدام البرمجيّات المغلقة لأغراض بحثيّة ودفع جميع تكاليف إتفاقيّات الحقوق تراخيص الإستخدام طالما كانت أهداف البحث متعلّقةً بشكل مباشر بهذه البرمجيّات.

الفقرة 7- شفافية الإستثناءات:

يجب الحفاظ على جميع الإستثناءات التي يتم إصدارها ونشرها عبر موقع الحكومة على الإنترنت.

يجب أن يحدّد تبرير منح الإستثناء المتطلبّات التشغيليّة التي يجب أن يوفّرها البرنامج.

الفقرة 8- الرخص الإستثنائية

إذا لم تتمكّن إحدى المؤسسات الحكوميّة من تلبية متطلباتها باستخدام البرمجيّات المحدّدة في الفقرة 2 من هذا القانون يرخّص لها الحصول على برنامج مغلق لحفظ أو معالجة البيانات التي يجب حفظها، يتوجّب على هذه المؤسسة نشر تقرير في موقع الحكومة على الإنترنت لتوضيح المخاطر المترافقة مع استخدام هذا البرنامج لهدف محدّد.

تعفى الرخص الإستثنائية الممنوحة للمؤسسات الحكوميّة ذات الطابع الأمني والدفاع الوطني من الشرط المذكور آنفاً.

الفقرة 9- المسؤوليات

تقع مسؤوليّة تنفيذ هذا القانون على عاتق أعلى سلطة إدارية في الحكومة إضافةً إلى دوائر تقنيّة المعلومات في كل المؤسسات الحكوميّة.

الفقرة 10- تطبيق القانون

ستقوم الجهات التنفيذيّة في الحكومة خلال 180 يوماً كحد أقصى بتحديد الشروط والمواعيد والأساليب الضابطة للتحول من الحالة الراهنة إلى الحالة المتوافقة مع شروط هذا القانون، وستقيّد في هذا السياق جميع العقود والمفاوضات المستقبليّة المتعلقة بشراء البرمجيّات.

كما ستقوم بتوجيه عمليات التحوّل من أنظمة البرمجيّات المغلقة إلى البرمجيّات المفتوحة في جميع الحالات التي تتطلب ظروفها ذلك.

الفقرة 11- المصطلحات

برنامج: أيّة مجموعة من التعليمات المستخدمة من قبل أنظمة معالجة البيانات الرقميّة لتنفيذ مهمّة محدّدة أو حلّ مشكلة ما.

تنفيذ أو تشغيل البرنامج: فعل استخدام هذا البرنامج على أيّ نظام لمعالجة البيانات الرقميّة لتنفيذ مهمّة ما.

المستخدم: شخص حقيقي أو إعتباري يستخدم البرنامج.

الشيفرة المصدريّة (أو برنامج الشيفرة المصدريّة): المجموعة الكاملة من التعليمات والملفات الرقميّة المصدريّة التي بنيت من قبل الأشخاص الذين قاموا ببرمجتها، إضافةً إلى جميع الملفات الرقميّة المساعدة كجداول البيانات والصور والمواصفات وملفات التوثيق وأيّة عناصر ضروريّة أخرى لبناء البرنامج التنفيذي. كاستثناء يمكن عدم شمل جميع الأدوات المتوفرّة عادة كبرمجيّات حرّة كبرامج التشغيل والمكتبات.

البرنامج الحر: وهو البرنامج الذي يضمن للمستخدم دون أيّة تكاليف إضافّيةً ما يلي:

- حريّة استخدام البرنامج لأي غرض.
- حريّة الحصول على الشيفرة المصدريّة للبرنامج.
- حريّة تفحص آلية عمل البرنامج.
- حريّة استخدام آليات عمل البرنامج أو أيّة أجزاء منه لتطويعها في تلبيّة متطلبات المستخدم.
- حريّة نسخ وإعادة توزيع البرنامج.
- حريّة تعديل البرنامج وحريّة توزيع هذه التعديلات ضمن نفس إتفاقيّة ترخيص البرنامج الأصلي.

البرنامج المغلق (البرنامج غير الحر): هو البرامج الذي لا يلبّي جميع المتطلبات المذكورة في التعريف السابق.

الصيغة المفتوحة: أيّ شكل من البيانات المرمّزة رقميّاً والمتوافقة مع المعايير القياسية إضافة إلى ما يلي:

- الوثائق الفنيّة المتعلقة به متاحة للعموم.
- الشيفرة المصدريّة لمثال مرجعيّ واحد على الأقل متاحة للعموم.
- لا توجد أيّة قيود على تطوير برمجيّات تستخدم هذا الأسلوب لحفظ أو إرسال أو استقبال أو الوصول إلى البيانات.

يبلّغ رئيس الجمهوريّة لإعلانه

ليما في التاسع من نيسان 2002

أسئلة عليك إجابتها

مع الإنتشار المتزايد للإنترنت وتحوّل العالم باتجاه المجتمع الرقمي يتزايد الحديث عن الحكومة الإلكترونيّة وما تستطيع تقديمه من رفع لكفاءة وفاعلية المؤسسات الحكوميّة، وقد أصبحت الحكومات اليوم أكثر من أيّ يوم مضى مضطرةً لاتخاذ قرارات حاسمة فيما يتعلق بالحلول التي ستعتمدها لبناء بنيتها المعلوماتيّة، ولا بدّ في هذا السياق من تقييم ومراجعة جميع الحلول المتاحة لتحديد مكامن الضعف والقوّة في كل منها واعتماد أكثرها ملاءمة للظروف المحليّة، ولتقييم استخدام البرمجيّات مفتوحة المصدر في المؤسسات الحكوميّة علينا الإجابة على الأسئلة التالية:

- ما هي متطلبات التفاعل بين المواطنين والمؤسسات الحكوميّة؟
- ما هي أهميّة إعتماد الصيغ المعياريّة لحفظ المعلومات في المؤسسات الحكوميّة؟
- ما هي التكاليف المباشرة وغير المباشرة المترتبّة على اعتماد كل من الحلول المتاحة؟
- ما هي مستويات الأمن المطلوبة لحماية المعلومات الحكوميّة؟
- ما هو الدور الذي تستطيع الحكومة لعبه لتطوير صناعة تقنيّة المعلومات المحليّة؟
- كيف تسهم البرمجيّات مفتوحة المصدر في تحرير الحكومات من هيمنة الشركات الإحتكاريّة؟

الفصل التاسع

التحوّل إلى المصادر المفتوحة

مقدّمة

سنحاول في هذا الفصل تزويد مدراء المعلوماتية ومتخذي القرار بالخطوط العريضة التي ستساعدهم في التخطيط للإنتقال إلى برمجيات المصادر المفتوحة.

هناك العديد من المبررات التي تدفع الشركات والمؤسسات للتحوّل من البرمجيات المغلقة إلى البرمجيات المفتوحة، منها على سبيل المثال: تزايد الحاجة إلى معايير ومقاييس موحدّة لتقنية المعلومات، مستوى الأمن المتطوّر الذي توفره برمجيات المصادر المفتوحة، التخلّص من مسلسل التطوير الإجبارية، وتكاليف البرمجيات . مما يؤدي إلى تخفيض نفقات تقنية المعلومات في هذه المؤسسات.

ينبغي قبل البدء بعملية الإنتقال إلى برمجيات المصادر المفتوحة التأكد من النقاط التالية:

- توفّر التفهم الكامل لأسباب التحوّل قبل البدء.
- التحقق من توفّر دعم فعّال لعملية التحوّل من قبل عناصر تقنية المعلومات إضافة إلى المستخدمين.
- وجود إرادة حقيقية لإنجاز التحول على مستويات الإدارة العليا في المؤسسة.
- بناء المهارات الملائمة في مجال المصادر المفتوحة.
- البدء بالأنظمة غير الحيوية.
- قابلية إدارة كل مرحلة من مراحل التحوّل.

تشكّل عملية التحول فرصة هامّة لإعادة تقييم أنظمة المعلومات في المؤسسة وإعادة هيكليتها إذا دعت الحاجة بما يجعلها تتلاءم مع متطلبات المؤسسة، يجب الإجابة على الأسئلة التالية أثناء عملية التقييم:

- كيف ستضمن التوافقية بين الأنظمة المختلفة؟
- كيف سيتم توفير الأمن الملائم للمعلومات؟
- كيف ستبنى الأنظمة الجديدة لتكون سهلة الإدارة؟

تشتهر البرمجيات مفتوحة المصدر بإستخدامها الواسع النطاق في عالم المخدّمات، مما يمكّن من تحويل المخدّمات إلى المصادر المفتوحة دون التأثير على المستخدمين، ولذلك عادةً ما يتم البدء بالمخدّمات كخطوة أولى في عملية التحوّل.

أما التحوّل إلى المصادر المفتوحة على سطح المكتب، وبالرغم من أنه يشكل الوفر الأكبر في تكاليف تقنيات المعلومات لأي مؤسسة، فيتطلب الدراسة بعناية وتأن للتأكد من توفير التوافقة الملائمة بين جميع التطبيقات البرمجية.

تعتمد الخطوات التفصيلية لعملية التحوّل بشكل أساسي على حجم المؤسسة ومقدار إعتمادها على تقنية المعلومات، فقد تتطلب هذه العملية بناء بيئة مختلطة تعمل ضمنها البرمجيات مفتوحة المصدر جنباً إلى جنب مع البرمجيات المغلقة (خاصةً عندما يكون حجم المؤسسة كبيراً لدرجة تزداد معها المدّة الزمنية اللازمة للتحوّل بشكل كبير)، وقد يتسبب في ذلك عدم توفّر بدائل ملائمة مفتوحة المصدر لبرمجيات هامّة تستخدم في المؤسسة.

يجب التأكد من أن أية قرارات سيتم اتخاذها في الوقت الراهن (حتى إذا لم تكن على صلة مباشرة بعملية التحوّل) لن تتسبب في توريط المؤسسة باستخدام تقنيات مغلقة أو غير قياسية.

البرمجيات مفتوحة المصدر مشاكسة، فهي عند دخولها مؤسسةً ما تقلب المفاهيم رأساً على عقب، وتغير جذرياً أسلوب العمل المتعلق بتقنية المعلومات في هذه المؤسسة، فهي تحوّل صناعة تقنية المعلومات من صناعة تعتمد على المنتج إلى صناعة خدمية، هذه البرمجيات مجانية، ولكن لا بدّ من البحث عن دعم لهذه البرمجيات. هناك العديد من الشركات والمجموعات التي توفّر الدعم الفنّي لبرمجيات المصادر المفتوحة، ولكن إذا كنت من الناس الذين يبحثون دوماً عن مشجب ليلومونه عند حدوث مشكلة ما، فمن المؤكد أن البرمجيات مفتوحة المصدر لن تلبّي متطلباتك. لا بدّ من تفهم آليات عمل المصادر المفتوحة بشكل جيّد قبل اتخاذ أي قرار يتعلق بعملية التحوّل.

منهجية العمل

لا بدّ أن تتوفر في أية عملية تحوّل النقاط التالية:

1. مرحلة جمع المعلومات وتوصيف المشروع، وتشمل:
   a. توصيف الحالة والظروف الراهنة، والتي تضم على سبيل المثال:
      i. بنية الأنظمة الحالية،
      ii. التطبيقات البرمجية المستخدمة والبيانات المرتبطة بها،
      iii. المعايير والبروتوكولات المستخدمة،

iv. التجهيزات،

v. البيئة الفيزيائية: كبنية الشبكة والموقع..

vi. المتطلبات البشرية كالمهارات اللازمة والمتطلبات اللغويّة.

b. تحديد الغاية من التحوّل وتوصيف الحالة المطلوبة بالتفصيل.

c. توصيف مراحل الإنتقال من الحالة الراهنة إلى الحالة المطلوبة.

2. تحديد مبررات التحوّل وتكاليفه.
3. تصميم مرحلة تجريبية أو أكثر للتحقق من الخطّة الموضوعة وإثبات المبررات.
4. تنفيذ الخطّة الموضوعة.
5. مقارنة التنفيذ الفعلي مع الخطة الموضوعة.

إن التنوع الهائل لتفاصيل كل من الحالة الراهنة والحالة المطلوبة بسبب اختلاف متطلبّات المؤسسات والشركات المختلفة يجعل من شبه المستحيل علينا أن نتمكن من تغطية جميع سيناريوهات التحوّل الممكنة، ولذلك فإن المعلومات الواردة في هذا الفصل تقدّم الخطوط العريضة التي ستساعد المدراء والفنيين ومتخذي القرار على تصميم خطة التحوّل الخاصة التي تلبّي متطلباتهم المختلفة، ويجب استخدام هذه المعلومات لنقطة بدية لعملية التحول، ولا ينبغي أن نتوقع منها أن توفّر جواباً لكل الأسئلة الممكنة.

سنعتبر أن هدف التحوّل هو الوصول إلى بيئة لا تحتوي سوى البرمجيات مفتوحة المصدر إن أمكن، ولكننا سنأخذ بالحسبان الحالات الخاصّة التي ينبغي فيها الإحتفاظ ببعض البرمجيات المغلقة لسبب ما أو لآخر.

لمحة عن عملية التحوّل

إن المتطلبات الأساسية لعملية التحوّل من البرمجيات المغلقة إلى البرمجيات مفتوحة المصدر لا تختلف عن متطلبات أية عملية تحوّل أخرى (التحوّل من ويندوز NT إلى ويندوز 2000). فحتى عندما يكون التحوّل بين بيئتين مختلفين تنتجهما نفس الشركة لا يمكن ضمان التوافق المطلق بين صيغ الملفات على سبيل المثال، ولا بد من إجراء الإختبارات اللازمة قبل القيام بأية عملية تحوّل واسعة النطاق، ولا بد من توفّر القدر الملائم من التخطيط والإدارة.

تشمل عملية التحول الخطوات التالية، والتي يمكن القيام ببعضها على التوازي:

قد تتطلب عملية التحوّل إلى المصادر المفتوحة إجراء بعض التعديلات على أنظمة المعلومات الراهنة قبل التمكّن من الإنتقال إلى أنظمة المصادر المفتوحة، ولذلك ينبغي أن تبني جميع الشركات والمؤسسات والتي قد لا ترغب بالقيام بعملية التحوّل في الوقت الراهن ولكنها تريد الحفاظ على هذا الخيار في حال أرادت ذلك في المستقبل بناها التحتية على أنظمة معيارية وقياسية دون التوجّه نحو التقنيات الخاصّة بمورّد معين.

1. شكّل فريقاً يملك المهارات الفنية وزوّده بدعم الإدارة: لا بد من توفّر الدعم الكافي من قبل الإدارة لمواجهة مقاومة التغيير في المؤسسة بإتجاه المصادر المفتوحة نتيجة الخوف من المجهول.
2. إدرس الحالة المطلوبة بشكل جيد، وقم بمراجعة جميع الخيارات المتاحة. قد يتطلب ذلك تدريب الموظفين في المؤسسة أو توظيف مهارات جديدة أو الإستعانة بالخبراء والإستشاريين، وبما أن ذلك سيتطلب بعض المصاريف المبدئية لا بدّ من

توفر الدعم الملائم من قبل الإدارة لتسهيل هذه المهمة. تجنّب الفهم المغلوط الشائع في أن الحصول على البرمجيات مفتوحة المصدر واستخدامها لا يترافق مع أية تكاليف.

3. تشكّل عملية التحوّل فرصةُ ذهبية لمراجعة وتقييم البنية الحالية لأنظمة المعلومات والتطبيقات البرمجية.
4. من المهم جداً أن يتوفّر لديك التفهم الكافي لفلسفة المصادر المفتوحة، عليك إعتبار النقاط الأساسية التالية قبل إتخاذ أية قرارات تتعلق بعملية التحوّل:
   a. توفّر الفهم الملائم لإتفاقيات ترخيص المصادر المفتوحة، خاصة إذا ما كانت المؤسسة ستقوم بتعديل وغعادة توزيع بعض هذه البرمجيات.
   b. عند وجود عدة بدائل ملائمة لتطبيق ما (كوجود أربعة محررات نصوص مفتوحة المصدر) يجب دراسة ميزات وعيوب كل من هذه البدائل.
   c. يجب إدراك الفروقات بين التوزيعات المختلفة، بعض التوزيعات تنتج من قبل شركات تجارية توفّر الدعم الفني والخدمات لها، في حين تقوم بعض مجموعات المبرمجين بإصدار توزيعات أخرى، ولا بد من أخذ جميع هذه الإختلافات بعين الإعتبار قبل إختيار التوزيعة التي ستقوم باستخدامها.
   d. يجب تحديد متطلبات الدعم الفني للمؤسسة، يمكن الحصول على الدعم الفني من مطوري البرمجيات أو منتجي التوزيعات لقاء مقابل مادّي، كما توفّر بعض الشركات خدمات الدعم الفني لبرمجيات المصادر المفتوحة بسبب توفّر الشيفرة المصدرية لهذه البرمجيات.

   وهذا فرق جوهري بين برمجيات المصادر المفتوحة وبرمجيات المصادر المغلقة والتي لا يتمكن مستخدمها من الحصول على الدعم الفني الملائم

إلا من الشركات المنتجة لهذه البرمجيات والتي وحدها تملك الحق في الوصول إلى الشيفرة المصدرية، والذي قد يشكل تهديداً في حال أفلست الشركة المنتجة لهذه البرمجيات أو توقفت عن العمل دون تحرير الشيفرة المصدرية لمنتجاتها، وقد سبق أن حدث ذلك مرات عدة.

عندما لا تتوفر خدمات الدعم الفني المذكورة (وهي حالة نادرة جداً) يمكن الإستعانة بقوائم التراسل الخاصة بهذه البرمجيات حيث يتم طلب المساعدة من المستخدمين أو المطورين الخبراء، ويعتبر وجود قائمة تراسل فعّالة لأي برنامج مفتوح المصدر نقطة أساسية لإختياره واستخدامه.

5. قم بتوصيف الحالة الراهنة، هذه المعلومات لن تستخدم أثناء عملية التحول وحسب، بل ستشكل مورداً هاماً عند تحديد تكاليف عملية التحوّل وإيجاد المبررات الدافعة لها. تأكد من توثيق ما يلي:
   a. لكل تطبيق برمجي مستخدم حالياً:
      i. اسم البرنامج، الإصدار واسم الشخص المسؤول عنه.
      ii. عدد المستخدمين لهذا البرنامج.
      iii. ما هو نظام التشغيل الذي يعمل ضمنه هذا البرنامج حالياً، وما هي أنظمة التشغيل الأخرى التي يتمكن البرنامج من العمل ضمنها.
      iv. ما هي البرمجيات الأخرى المطلوبة لعمل هذا البرنامج على كل من المخدّم والزبون.
      v. ما هي متطلبات التجهيزات الخاصة لهذا البرنامج (هل يحتاج تشغيل البرنامج إلى تجهيزات خاصة أو غير قياسية؟).

vi. ما هي بروتوكولات الإتصال التي يستخدمها هذا البرنامج للتخاطب مع البرمجيات الأخرى.

vii. ما هي صيغ الملفات التي يستخدمها هذا البرنامج.

viii. ما هي المتطلبات اللغويّة لهذا البرنامج، قد يتطلب البرنامج دعم عدّة لغات أو عملات على سبيل المثال.

b. متطلبات البيانات، والتي يجب تحديدها بالتفصيل، كملفات النصوص والجداول الحسابية وملفات الصوت والصورة وقواعد البيانات.

i. ما هي متطلبات التفاعل مع أنظمة أو مستخدمين خارج نطاق المؤسسة.

ii. ما هي متطلبات الحفاظ على البيانات وإمكانية معالجتها في المستقبل؟ هل هناك كم من البيانات الحالية التي يجب توفير الدعم لها واستخدامها، وفي هذه الحال، هل تتطلب هذه البيانات برمجيات معينةً لمعالجتها؟

قم بتقسيم البيانات إلى الأنواع التالية:

1. البيانات غير الضرورية والتي يمكن الإستغناء عنها، قم بالتخلّص من هذه البيانات.

2. البيانات التي بجي الحفاظ عليها وهي موجودة حالياً بصيغ قياسية (كملفات PDF) أو يمكن تحويلها بسهولة إلى صيغ قياسية، ينبغي هنا تحديد تكاليف هذا التحويل.

3. البيانات التي يجب الحفاظ عليها ولكنها موجودة حالياً بصيغ غير قياسية ولا يمكن تحويلها بسهولة إلى صيغ

قياسية. قد يتطلب الحفاظ على هذه البيانات الإحتفاظ بالبرمجيات المغلقة المرتبطة بها لإتاحة التعامل معها. ينبغي تحديد تكاليف هذه البرمجيات المغلقة وعدد النسخ التي يجب توفيرها (والذي يمكن تحيدها تبعاً للحاجة إلى التعامل مع هذه البيانات). فإذا كانت هذه البيانات نادرة الإستخدام يمكن عندها الإحتفاظ بنسخة واحدة فقط من البرنامج المغلق، وقد يتطلب هذا الخيار أيضاً توفير بعض التجهيزات الخاصة لتشغيل هذه البرمجيات.

c. متطلبات أمن المعلومات:

i. ما هو النظام المستخدم حالياً لتحديد أسماء المستخدمين وكلمات السر؟ هل هناك قواعد محددة لأسماء المستخدمين؟ ما هي سياسة تعديل كلمة السر؟

ii. هل هناك أنظمة ذات متطلبات أمنية غير اسم المستخدم وكلمة السر؟

iii. هل هناك أية سياسات خاصة بأمن المعلومات في المؤسسة؟ علي سبيل المثال، هل هناك أية قيود على استخدام الإنترنت والبريد الإلكتروني؟

iv. هل هناك أية متطلبات أمنية تستوجب توفّر تجهيزات أو برمجيات خاصّة؟

6. قم بإعداد تقرير متكامل لخطة التحوّل يتضمن المعلومات التي قمت بجمعها، يجب أن يحتوي هذا التقرير على الأجزاء التالية:

a. تكاليف البنية الراهنة على مدىً معقول من الزمن (5 سنوات مثلاً).
b. تكاليف البنية البديلة وتكاليف عملية التحوّل على نفس الفترة الزمنية.
c. نقاط الضعف والقوّة في البنية الراهنة والخيارات البديلة المتعددة.

7. تشاور مع المستخدمين، ناقش معهم أسباب ومبررات عملية التحول وتأثيراتها المباشرة عليهم. تفهّم مشاكلهم واعتراضاتهم وتعامل معها بجديّة، شجّعهم على التعامل مع التقنية الجديدة، فكلما استعد المستخدمون للتعامل مع البنية الجيدة كلما كانت عملية التحوّل ذاتها أكثر سهولةً وسلاسةً.

في حال كانت المؤسسة كبيرة من الممكن تجهيز مركز خاص بالخدمات للإجابة على استفسارات المستخدمين، قد يتحوّل هذا المركز بعد الإنتهاء من عملية التحوّل كمركز للدعم الفنّي. قم ببناء موقع خاص ضمن شبكة الإنترانت الخاصة بالمؤسسة وضمنّه معلومات عن أهداف ومبررات وسير عملية التحوّل، يجب أن يتمكن المستخدمون أنفسهم من تعديل هذا الموقع، فمن الأهمية بمكان منح المستخدمين الإحساس بأنهم مشمولون بهذا التغيير، أو أنهم هم أنفسهم جزء من هذا التغيير، سيساعد هذا الموقع موظفي مركز الدعم الفنّي في التعرف على أنواع المشاكل التي يواجهها المستخدمون.

8. بعد إعداد خطّة العمل، إبدأ بتنفيذ المراحل التجريبية على نطاق ضيّق وبعدد محدود من المستخدمين، ستوفر لك هذه المرحلة ما يلي:

a. المزيد من المعلومات التي ستستخدم لتدقيق التكاليف المقدّرة للمشروع.
b. التعرف على ردّ فعل المستخدمين والتي ستسهل تعريف المستخدمين الآخرين بالأنظمة الجديدة.
c. التحقق من فاعلية البنية المصممة وخطة العمل وتعديلها إذا اقتضت الحاجة.

9. حدّد السرعة المطلوبة لإنجاز التحوّل بعد البدء. أمامك عدّة خيارات:

a. التغيير الشامل مرّة واحدة: حيث يتم نقل جميع المستخدمين من النظام القديم إلى النظام الجديد في يومٍ واحد، ويعني عملياً تنسيق التغيير ليتم في أحد أيام الإجازات أو العطل الرسمية. من ميزات هذا الخيار التخلص من ضرورة توفير بيئتين مختلفتين للمستخدمين في نفس الوقت وضرورة تنقّل المستخدمين بين النظامين الجديد والقديم، أما المساوئ فتتضمن إحتمالاتٍ أكبر لمواجهة مشاكل أثناء التحوّل والمتطلبات الكبيرة للموارد اللاومة لإتمام عملية التحوّل. ويعتبر هذا الخيار صائباً في حالة الشركات والمؤسسات الصغيرة فقط.

حاول قدر الإمكان تجنّب التغير الشامل لأنه ينطوي على الكثير من المتغيرات التي يجب إدارتها والتحكّم فيها مما يتسبب غالباً في إفشال العملية برمّتها، وفي هذه الحال لا ينبغي توجيه اللوم إلى برمجيات المصادر المفتوحة، بل إلى عملية إدارة التغيير.

b. التحوّل التدريجي ضمن مجموعات: حيث يتم نق المستخدمين من النظام القديم إلى الجديد في مجموعات. ويتم على الأغلب نقل الوحدات الإدراية مع بعضها لتقليل تأثيرات ومشاكل مشاركة البيانات والعمل الجماعي لكل من هذه الوحدات. يمكن إدارة الموارد وتجاوز العقبات في هذا الخيار عبر التحكّم بأحجام المجموعات.

c. التحوّل التدريجي على مستوى المستخدم: وهو يشبه إلى حدّ كبير التحوّل التدريجي ضمن مجموعات ولكن حجم المجموعة هنا شخص واحد. يتميز هذا الخيار بمتطلباته البسيطة ولكنه غير ملائمٍ للشركات

والمؤسسات الكبيرة. يمكن إستخدام هذا الخيار أثناء تنفيذ المرحلة التجريبية.

من الممكن أن تتطلب عملية التحوّل تشغيل النظامين القديم والجديد جنباً إلى جنب لفترة من الزمن، لذلك من الضروري أن تحتوي خطة العمل إستراتيجية واضحة للمرحلة الإنتقالية تمكّن النظامين القديم والجديد من العمل معاً مما يسمح باستمرار سير العمل خلال هذه المرحلة الإنتقالية. قد يستغرق نقل الحاسوب الأخير مدةً طويلةً جداً من الزمن (وقد لا يحدث على الإطلاق!) لذلك يعتبر تفاعل النظامين فائق الأهمية.

10. نفّذ خطّة العمل حتى يتم التحوّل ضمن المؤسسة بأكملها. قد يتطلب ذلك المزيد من التدريب للمستخدمين والفنيين.
11. تابع آراء المستخدمين وعالج أية مشكلةٍ حال ظهورها. قد يحتاج بعض المستخدمين متطلباتٍ خاصّة لا تظهر مقدّماً أو أثناء المراحل التجريبية، تأكد من وجود الموارد الكافية لتلبية هذه المتطلبات بعد عملية التحوّل.

العوامل البشرية

لا بدّ لضمان نجاح عملية التحوّل أن تتم استشارة جميع الموظفين وإعلامهم بأهداف ومبررات التحوّل للحصول على دعمهم ومساندتهم لخطة العمل خلال جميع مراحلها.
من الممكن مثلاً بناء موقع خاص ضمن شبكة الإنترانت الخاصّة بالمؤسسة لنشر المعلومات المتعلقة بالمشروع والحصول على تعليقات وآراء المستخدمين.

من العوامل الهامّة الأخرى والمتعلقة بالمستخدمين موضوع التدريب، ففي حين قد تخّير بعض المؤسسات موظفيها في حضّور فصول التدريب، قد تجبر مؤسسات أخرى الموظفين على الحضور، يتعلق هذا الخيار بطبيعة العمل ضمن المؤسسة وبموضوع الدورات التدريبية ذاته. معظم المعلومات المتوفرة عن البرمجيات مفتوحة المصدر بما فيها كتيبات التدريب وملفات المساعدة مكتوبة باللغة الإنكليزية فقط، مما قد يتسبب في بعض العقبات أمام الموظفين الذين لا يجيدون اللغة الإنكليزية، لا بدّ من أخذ هذه النقطة بعين الإعتبار ومساعدة هؤلاء الموظفين على تجاوز هذه العقبات. قد تتم ترجمة بعض هذه المستندات إلى اللغة العربية وإعتبارها إحدى تكاليف عملية التحوّل.، ولكننا سنواجه بعد ذلك مشكلة تعديل هذه المستندات وتطويرها.

تقدم واجهات البرمجيات المفتوحة المصدر الرسومية (وبشكل خاص بيئتي KDE و Gnome) خيارات واسعة من اللغات المختلفة، ولكن بعض هذه اللغات قد لا تكون مكتملةً بعد مما يجعل بعض التعليمات والأوامر وملفات المساعدة تظهر باللغة الإنكليزية. وقد لا تحتوي جميع البرمجيات المستخدمة على دعمٍ كاملٍ للإعدادات المحليّ. جميع هذه العوامل تتغير بسرعة كبيرة مما يبشّر بإمكانية تشغيل برمجيات المصادر المفتوحة بأية لغةٍ في المستقبل القريب.

يجب أن تأخذ خطة العمل بعين الإعتبار ردود الفعل التقليدية لأي عملية تغيير والتخطيط لطرائق معالجة كلٍ منها:

1. الخوف من المجهول:

سيكون إستخدام برمجيات المصادر المفتوحة جديداً تماماً لكثيرٍ من المستخدمين، وسيتسبب الخوف الطبيعي من المجهول في مقاومة هؤلاء الأشخاص لبرمجيات

المصادر المفتوحة لأنها جديدة بالنسبة لهم، فالمرء أسير ما يعلم.

بعض المستخدمين سيبدون إهتماماً بالفكرة الجديدة ولن يمانعوا في تجربتها بكل سعة، يجب أن تبدأ عملية التحوّل عند هذا النوع من الأشخاص. أظهرت التجربة أن المستخدمن وبمجرد أن ينهار حاجز الوهم بينهم وبين برمجيات المصادر المفتوحة لن يجدوها مختلفة بشدّة عن البرمجيات المغلقة بل سيعتادون عليها ويجدوها أكثر متعةً وإثارةً. ولذلك فإن هذه المجموعة من المستخدمين سينتقلون إلى البرمجيات مفتوحة المصدر بحماس، وهم عادةً من يقدّم الآراء والتعليقات المفيدة.

يمكن إستخدام المجموعة الأولى من المستخدمين في المراحل التجريبية، وبعد نجاح هذه المراحل وحصول المستخدمين على الخبرات والمهارات الملائمة سيقومون بنقل هذه الخبرات والنجاحات إلى زملائهم. على أية حال فإن المستخدمين في المرحلة الثانية سيتطلبون المزيد من خدمات الدعم كمركز الدعم الفنّي، الإنترانت وزملاءهم الأكثر خبرةً ودرايةً بالموضوع الجديد.

2. تدعيم السيرة الذاتية:

قد يعتبر المستخدمون أو مدراء الأنظمة أن استخدامهم لبرمجيات تختلف عن تلك "الأكثر شهرةً في السوق" قد يؤثر سلباً على مهاراتهم ويؤذي قدراتهم على تطوير حياتهم المهنية. هذه المشكلة معقدّة للغاية وتتطلب الكثير من الإدارة الحذقة. وحتى يتسع نطاق استخدام البرمجيات مفتوحة المصدر فإن هذه المشكلة ستطفو على سطح أية عملية تحوّل.

3. المعلومة هي مصدر القوة الأساسي في عصر المعرفة:
يملك الموظفون العارفون بتفاصيل الأنظمة الحالية وإعداداتها قدراً من القوة لن يتنازلوا عنه بسهولة، خاصةً إذا كانت برمجيات المصادر المفتوحة مختلفةً كليةً عن الأنظمة المستخدمة حالياً، وهذه مشكلة أخرى تتطلب الإدارة بعناية لأن هؤلاء الأشخاص يلعبون دوراً أساسياً في تشغيل الأنظمة الحالية. قد يتوجب إيلاء هؤلاء الموظفين الأولوية في التدريب على الأنظمة الجديدة لطمأنتهم على الحفاظ على مواقعهم في العمل.

الإنتقال المرحلي

تعمل معظم البرمجيات مفتوحة المصدر ضمن أنظمة التشغيل المغلقة بالإضافة إلى أنظمة التشغيل المفتوحة مما يوفّر إمكانية تعريف المستخدمين بهذه البرمجيات دون تغيير بيئة العمل بأكملها. يمكن على سبيل المثال تشغيل OpenOffice و Apache وMozilla ضمن ويندوز لإستبدال كلٍ من Office و Internet Information Server و Internet Explorer على التوالي. يتيح هذا النمط من التحوّل عدا عن كونه أكثر طرق التحوّل سلاسةً تخطيط وتنظيم عمليات التدريب بما يتلاءم مع التجارب الفعليّة التي ستواجه المستخدمين. كما تسهّل عمليات تحويل صيغ الملفات بالحفاظ على البرمجيات القديمة بشكل مؤقت جنباً إلى جنب مع البرمجيات مفتوحة المصدر.

سيتحدّد في هذا النمط من التحوّل خيار البرمجيات مفتوحة المصدر المستخدمة بالبرمجيات التي تستطيع العمل ضمن الأنظمة الحاليّة. فقد يتم إختيار متصفح الوب Konqueror ولكنه لا يعمل ضمن ويندوز لذلك لا بدّ من استخدام المتصفح Mozilla.

الأسهل فالأصعب

أثناء تنفيذ خطّة التحوّل ينبغي أن يبدأ تنفيذ الخطوات التي تنطوي على أقل قدرٍ من التفاعل مع المستخدمين أولاُ، ولذلك يجب نقل المخدّمات أولاً، معظم التعديلات التي ستجرى على المخدّمات متوافقة بطبيعتها مع أنظمة المستخدمين الحاليّة مما يقلل من الآثار السلبية على أنظمة المستخدمين أثناء عملية التحوّل.

على سبيل المثال قد تكون مخدّمات DNS وDHCP عرضةً للتطوير إلى برمجيات المصادر المفتوحة دون أن يؤثر هذا التحوّل على أداء هذه المخدّمات ضمن الشبكة.

فكّر بالمستقبل

إن القيام ببعض الخطوات والإجراءات البسيطة اليوم سيجعل من أية عملية تحوّل مستقبلية في حال تمّ إتخاذ قرار بتنفيذها أمراً أكثر سهولةً، فيما يلي بعض النقاط التي يجب أخذها بعين الإعتبار في هذا السياق:

1. تأكد من أن جميع البرمجيات المعتمدة على الوب والتي يتم تطويرها إما ضمن الشركة أو عبر شركات أخرى مصممة بطريقة تتيح عرض نتاجها صمن أي متصفح للوب (وبشكل خاص المتصفحات مفتوحة المصدر). هذا القرار سيحرر المؤسسة من الحاجة إلى برنامج معين دون غيره. تتوفر بعض البرمجيات التي قد تساعد على التحقق من هذا الشرط (weblint).

2. حاول التخلّص من أية نصوص برمجية مضمنّة في المستندات والجداول الحسابية عبر البحث عن بدائل تلبّي نفس المتطلبات. يعتبر هذا الخيار جيداً أيضاً للتخلّص من مشاكل الفيروسات التي تستخدم مثل هذه النصوص البرمجية، كما يساعد في تدعيم أمن المعلومات.
3. تأكد دوماً من استخدام صيغ قياسية للملفات، كملفات PDF.

يدور نقاش كبير حول ماهيّة صيغة الملفات PDF وما إذا كانت فعلاً قياسيةً أم لا، والحقيقة أن هذا النقاش يتمحور فعلياً حول ماهيّة الجهة التي تتحكم بهذه المعايير. وعملياً فإن صيغة PDF هي من صيغ الملفات القليلة المستخدمة على نطاق واسع في الوقت الراهن والتي تتمتع بتوصيفٍ دقيقٍ مفتوح يمكّن من إستخدامها دون أية قيود.

هناك العديد من المحاولات التي تهدف إلى بناء صيغ قياسية للملفات معتمدةً على لغة XML ولمن هذه المحاولات مازالت في بداياتها وستحتاج بعض الوقت حتى تتحول إلى صيغ قياسية واسعة الإستخدام.

تجنّب بشكل خاص استخدام صيغ الملفات غير القياسية للملفات التي لن يتم تعديلها (أي الملفات التي يمكن قراءتها فقط) من قبل المستلم، سيخفّض ذلك من مشاكل الفيروسات التي تستخدم الملفات المرسلة عبر البريد الإلكتروني وسيلةً أساسيةً لإنتشارها. إن استخدام صيغ ملفات غير قياسية سيؤدي إلى تقييد المؤسسة بمنتج هذه الصيغ لفترة من الزمن، وتحتوي صيغ الملفات غير القياسية عادةً معلوماتٍ مخفيةً مضمنّةً كنصوص تمّ حذفها من الملف الأساسي والتي قد تتسبب في حال إظهارها بالحرج للكثير من المؤسسات.

4. عند مشاركة الملفات مع الآخرين حاول قدر المستطاع استخدام أكثر صيغ الملفات إنتشاراً، استخدم مثلاً ملفات Word97 عوضاً عن ملفات Word2000 لأن ذلك قد يسهّل مهمة إيجاد برمجيات مفتوحة المصدر تستطيع التعامل مع هذه الملفات.
5. استخدم البروتوكولات القياسية المفتوحة دوماً، تعرف البروتوكولات القياسية المفتوحة بأنها البروتوكولات التي لا تشمل ضمن أي تسجيل ملكية ولها تطبيق في البرمجيات مفتوحة المصدر. هناك العديد من هذه البروتوكولات القياسية المفتوحة الخاصّة بكل دولة، مثل E-gif في إنكلترا و OSOSS في هولندا و SAGA في ألمانيا.
6. تأكد من أن جميع البرمجيات الجديدة مصممة للعمل ضمن أكثر من بيئة تشغيل، أي أنها مطورة باستخدام لغات برمجية معيارية مثل ANSI C، Java و Perl. تجنّب استخدام أية أدوات تطوير تتطلب وجود أية برمجيات أو مكتبات مغلقة أو تعمل ضمن نظام تشغيل واحد.
7. أبعد المستخدمين عن برمجيات البريد الإلكتروني المغلقة والتي تحتفظ بالرسائل بصيغ غير قياسية وتتخاطب مع المخدّم باستخدام بروتوكولات غير قياسية. تأكد من أن برمجيات البريد الإلكتروني تستخدم بروتوكولات معيارية مثل IMAP.

أسئلة عليك إجابتها

لا بد أن تفكر بشكل ما أو بآخر في أهمية المصادر المفتوحة وما قد توفره لمؤسستك أو شركتك من ميزات وفوائد، ولكن أي قرار ستتخذه في التحول إلى اعتماد هذه البرمجيات يجب أن يكون مستنداً إلى دراسة مستفيضة لواقع مؤسستك ومدى ملاءمة البرمجيات مفتوحة المصدر لهذا الواقع، وفي حال اتخذت قراراً بالتحول إلى المصادر المفتوحة فإن

نجاح تنفيذ هذا التحول سيتطلب أيضاً أخذ جميع العقبات والمشكلات التي قد تواجه هذه التنفيذ بعين الإعتبار، وبناء خطة التحول لتعكس جميع الحلول والخيارات الممكنة، عليك أن تجيب عن الأسئلة التالية للتحقق من إمكانية تنفيذ الخطة التي بنيتها:

- ما هي الفوائد التي ستجنيها المؤسسة من التحول إلى المصادر المفتوحة؟
- هل ستوفر الإدارة الدعم اللازم لضمان نجاح عملية التحول؟
- ماهي التكاليف المترافقة مع تنفيذ عملية التحول؟
- ما هي متطلبات التوافقية والتفاعل بين الأنظمة المختلفة (سواءٌ كانت مفتوحة أم مغلقة المصدر)؟
- ما هي العقبات التي قد تواجه تنفيذ خطة التحول؟ وهل تتضمن الخطة خطوات لتجاوز هذه العقبات؟
- ما هو مدى تفاعل المستخدمين في المؤسسة مع قرار التحول باتجاه المصادر المفتوحة؟
- ما هي متطلبات التدريب المترافقة مع عملية التحول؟
- ما هو الإطار الزمني لعملية التحول؟

الفصل العاشر
دراسة حالة في المصادر المفتوحة

مقدّمة

تحدّثنا في الجزء الأكبر من هذا الكتاب عن فلسفة البرمجيات مفتوحة المصدر وما تقدّمه هذه البرمجيات من ميزات وعوائد. وقد يتبادر إلى ذهنك الآن سؤال مفاده: حسناً، كل هذا الكلام رائع، أصبحت شبه مقتنعاً بالبرمجيات مفتوحة المصدر، ولكن ما الذي يتوجب عليّ عمله للإفادة من هذه البرمجيات في أعمالي؟

وللإجابة عن هذا السؤال قمت بإعداد هذا الفصل، والذي سنبتكر فيه شركةً تخيليةً اتخذت قراراً باعتماد المصادر المفتوحة، سنحاول متابعة كل الخطوات التي قامت بها هذه الشركة نحو اعتماد المصادر المفتوحة، بدءاً من مبررات هذا القرار وانتهاءً بالتحول الكامل إلى البرمجيات مفتوحة المصدر، آملاً أن يكون هذا المثال وسيلة ستساعد الكثيرين على المضي قدماً في اتخاذ قرار التحول إلى المصادر المفتوحة.

ملاحظة:
إن الشركة والأشخاص الوارد ذكرهم في هذا الفصل هم من نسج خيال المؤلف ولا علاقة لهم بالواقع، وأي تشابه في الوصف أو الأسماء ليس إلا محض صدفة.

الوضع الراهن في الشركة العربية

ترغب الشركة العربية (وهي شركة تجاريّة تعمل في مجال الإستيراد والتصدير وتوظف 75 موظفاً) في تطوير بنية شبكتها الحاسوبية، تتألف هذه الشبكة من 50 حاسوباً متصلةً مع بعضها عبر شبكة محلية تعمل بتقنية إيثرنت Ethernet بسرعة 100 ميغابت في الثانية وتتشارك في استثمار مخدّم ذي معالجين يقوم بتوفير خدمات البريد الإلكتروني وحفظ الملفات والنسخ الإحتياطي. ترتبط هذه الشبكة مع الإنترنت عبر موجّه Router بسرعة 2 ميغابت في الثانية.

تستخدم الشركة العربية على حواسيبها نظام التشغيل مايكروسوفت ويندوز 98 Microsoft Windows 98 وحزمة البرمجيات المكتبية مايكروسوفت أوفيس 2000 Microsoft Office 2000 ومتصفح الإنترنت Microsoft Internet Explorer 6.

أما المخدّم فيشغّل نسخة مايكرسوفت ويندوز Microsoft Windows NT ومخدّم البريد الإلكتروني Microsoft Exchange 5.5.

بعد اتخاذ قرار التطوير، كلّفت إدارة الشركة جابر (وهو مدير تقنية المعلومات في الشركة) بدراسة البدائل المتوفرة وتقديم تقرير يوضح فيه هذه البدائل لاتخاذ قرار باعتماد أحدها.

الحلول المتاحة

كان من السهل جداً على جابر أن يقوم بإعداد تقريره بسرعة فيما لو قرر اعتماد البرمجيات المغلقة، حيث أن الخيار الوحيد تقريباً كان اعتماد الإصدارات الجديدة من برمجيات مايكروسوفت، ولكنّ جابر (وكمعظم محترفي تقنية المعلومات) يحبّ العمل الإضافي وسهر

الليالي، وكان قد سمع مؤخراً عن البرمجيات مفتوحة المصدر وما يمكن أن تقدّمه للشركة، ولذلك فقد قرر جابر أن يقوم بإعداد تقرير يتضمن جميع الحلول المتاحة سواء كانت مغلقة أم مفتوحة المصدر.

أراد جابر أن يكون أي حل سيقدّمه في تقريره متكاملاً وقابلاً للتطبيق في الشركة العربية، وفي سبيل ذلك فقد بنى منهجية عمله على أساس أولوية تلبية المتطلبات الأساسية للشركة العربية ومن ثمّ تأتي الخدمات الإضافية.

تنوي الشركة العربية بناء موقع لها على الإنترنت، وترغب باستضافة هذا الموقع على المخدّم الموجود لديها، سيحتوي هذا الموقع على قواعد بيانات للزبائن يمكّنهم من التواصل مع الشركة عبر الموقع، أمن هذا الموقع فائق الحيوية نظراً لارتباطه مع قاعدة البيانات التي تحتوي على معلومات الزبائن.

متطلبات الشركة العربية

تحتاج الشركة العربية لضمان سير العمل إلى مجموعة من الخدمات التي يجب أن توفرها الحلول المقترحة، هذه الخدمات موضّحة في الجدول التالي:

البرنامج الذي يقوم بتوفيرها حالياً	إسم الخدمة
Microsoft Exchange 5.5	البريد الإلكتروني
Microsoft Windows NT 4	مشاركة الملفات
Microsoft SQL Server	قواعد البيانات
Microsoft ISA Server	ربط الإنترنت
Microsoft Office 2000	التطبيقات المكتبية
Microsoft Internet Explorer 6	تصفح الإنترنت
Microsoft Windows NT 4	DNS
Microsoft Windows NT 4	DHCP
Microsoft Windows NT 4	Directory Services
ArcServ	النسخ الإحتياطي

الحل الأول: البرمجيات المغلقة

قام جابر ببحث مطول على الإنترنت خلص بنتيجته إلى اختيار مجموعة البرمجيات التي تلبي متطلبات شركته الأساسية مع إعتبار التوافقية فيما بينها، وقام بتلخيص النتائج في الجدول التالي:

	إسم البرنامج	الإصدار
نظام التشغيل للمخدم	مايكروسوفت ويندوز Microsoft Windows	2000 Server
نظام التشغيل للحواسيب	مايكروسوفت ويندوز Microsoft Windows	XP
حزمة التطبيقات المكتبية	مايكروسوفت أوفيس Microsoft Office	XP
متصفح الإنترنت	إنترنت إكسبلورر Microsoft Internet Explorer	6
مخدّم البريد الإلكتروني	Microsoft Exchange Server	2000
مخدم الوب	Microsoft Internet Information Services	6
مخدّم قواعد البيانات	Microsoft SQL Server	2000
ربط الإنترنت	Microsoft ISA Server	2000
النسخ الإحتياطي	ArcServ	

هذه البرمجيات ستلبّي المتطلبات الأساسية للشركة العربية كما هو موضّح في الجدول التالي:

إسم الخدمة	البرنامج الذي سيقوم بتوفيرها في الحل المقترح	البرنامج الذي يقوم بتوفيرها حالياً
البريد الإلكتروني	Microsoft Exchange 2000	Microsoft Exchange 5.5
مشاركة الملفات	Microsoft Windows 2000 Server	Microsoft Windows NT 4
قواعد البيانات	Microsoft SQL Server	Microsoft SQL Server
ربط الإنترنت	Microsoft ISA Server	Microsoft ISA Server
التطبيقات المكتبية	Microsoft Office XP	Microsoft Office 20000
تصفح الإنترنت	Microsoft Internet Explorer 6	Microsoft Internet Explorer 6
DNS	Microsoft Windows 2000 Server	Microsoft Windows NT 4
DHCP	Microsoft Windows 2000 Server	Microsoft Windows NT 4
Directory Services	Microsoft Windows 2000 Server	Microsoft Windows NT 4
النسخ الإحتياطي	ArcServ	ArcServ

الحل الثاني: البرمجيات مفتوحة المصدر

في المقابل بحث جابر أيضاً عن البرمجيات مفتوحة المصدر التي تلبّي متطلبات شركته ولخّص النتائج في الجدول التالي:

	إسم البرنامج	**الإصدار**
نظام التشغيل للمخدم	Fedora	Core 1
نظام التشغيل للحواسيب	Fedora	Core 1
حزمة التطبيقات المكتبية	OpenOffice	1.1
متصفح الإنترنت	Mozilla	1.6
مخدّم البريد الإلكتروني	SendMail	8.12.11
مخدم الوب	Apache	1.3
مخدّم قواعد البيانات	MySQL	4.5
ربط الإنترنت	Squid	2.5
DNS	BIND	9.2.3
DHCP	DHCPD	3.5
Directory Services	OpenLDAP	2.2.6
النسخ الإحتياطي	Amenda	2.4.4

هذه البرمجيات ستلبّي المتطلبات الأساسية للشركة العربية كما هو موضّح في الجدول التالي:

البرنامج الذي يقوم بتوفيرها حالياً	البرنامج الذي سيقوم بتوفيرها في الحل المقترح	إسم الخدمة
Microsoft Exchange 5.5	SendMail	البريد الإلكتروني
Microsoft Windows NT 4	Fedora Core 1	مشاركة الملفات
Microsoft SQL Server	MySQL	قواعد البيانات
Microsoft ISA Server	Squid	ربط الإنترنت
Microsoft Office 20000	OpenOffice 1.1	التطبيقات المكتبية
Microsoft Internet Explorer 6	Mozilla 1.6	تصفح الإنترنت
Microsoft Windows NT 4	BIND	DNS
Microsoft Windows NT 4	DHCPD	DHCP
Microsoft Windows NT 4	OpenLDAP	Directory Services
ArcServ	Amenda	النسخ الإحتياطي

مقارنة التكاليف

لا بدّ أن تتوفر جميع المعلومات اللازمة لاتخاذ أي قرار من قبل الإدارة باعتماد أي من الحلين المقترحين، ومن أهم هذه المعلومات التكاليف المادّية المترتبة على اختيار أي من هذين الحلّين لتبرير هذا الإختيار أمام الإدارة المالية.

فيما يلي مقارنة التكاليف التي أعدّها جابر لكل من الحلين المقترحين:

الحل الأول: البرمجيات المغلقة

العدد	البيان	السعر الإفرادي (يورو)	السعر الإجمالي (يورو)
1	Microsoft Windows 2000 Server	4000	4000
50	Microsoft Windows XP	300	15000
50	Microsoft Office XP	500	25000
50	Microsoft Internet Explorer 6	0	0
1	Microsoft Exchange Server 2000	11500	11500
1	Microsoft Internet Information Services	0	0
1	Microsoft SQL Server 2000	20000	20000
1	Microsoft ISA Server	1500	1500
1	Veritas	900	900
الإجمالي لخمسين مستخدماً			77900
الإجمالي لكل مستخدم			1558

الحل الثاني: البرمجيات مفتوحة المصدر

العدد	البيان	السعر الإفرادي (يورو)	السعر الإجمالي (يورو)
1	Fedora Linux	0	0
50	Fedora Linux	0	0
50	OpenOffice	0	0
50	Mozilla	0	0
1	SendMail	0	0
1	Apache	0	0
1	MySQL	0	0
1	BIND	0	0
1	DHCPD	0	0
1	OpenLDAP	0	0
1	Squid	0	0
1	Amenda	0	0
الإجمالي لخمسين مستخدماً			0
الإجمالي لكل مستخدم			0

يتضح مما سبق أن الحل الذي يستخدم البرمجيات مفتوحة المصدر يتفوق على حل البرمجيات المغلقة بشكل كبير في الجانب الإقتصادي، حيث أنّه ينطوي على توفير هائل في تكاليف عملية التطوير.

المفاضلة بين الحلول المقترحة

لتسهيل عملية اتخاذ القرار فيما يتعلق باعتماد أحد الحلول المقترحة كان لا بدّ من تبيان الجوانب السلبية والإيجابية في كل من هذه الحلول وإيضاح مدى ملاءمة هذه الحلول لمتطلبات الشركة العربية.

تقدّم البرمجيات مفتوحة المصدر مستوىً متميزاً من الأداء والوثوقية، وهو ما يعتبر عاملاً فائق الأهمية بالنسبة للإدارة، فاستمرارية العمل هي من أهمّ الأولويات، وتوفر مستويات أعلى من الأداء والوثوقية سيرفع إنتاجية وكفاءة العمل في الشركة ويخفّض من هدر الوقت بسبب توقف البرمجيات عن العمل.

وتعتبر الإعتمادية من أهمّ ميزات البرمجيات مفتوحة المصدر، هذه الإعتمادية ستنعكس على تكاليف المتابعة والدعم الفنّي لهذه البرمجيات وبالتالي توفّر المبالغ المستثمرة في حلول أنظمة المعلومات.

ترغب الشركة العربية ببناء موقع لها على الإنترنت، هذا الموقع سيسهم في توفير خدمات إضافية للزبائن ويساعدهم على التفاعل مع الشكة بشكل أفضل، ولكنّ ذلك يتطلب تأمين معلومات الزبائن الموجودة ضمن قواعد البيانات المرتبطة مع الإنترنت وحمايتها من أية محاولات للتخريب أو السرقة. ولذلك فإن العامل الأمني سيلعب دوراً أساسياً في تحديد الحل الذي سيتم اعتماده.

تعتبر برمجيات المصادر المفتوحة أكثر الأنظمة البرمجية أمناً نتيجةً لطبيعة آليات تطوير البرمجيات مفتوحة المصدر، هذا المستوى من أمن المعلومات الذي تقدّمه البرمجيات مفتوحة المصدر سيشكل عاملاً حاسماً في تحديد الحل الذي ستقوم إدارة الشركة العربية باختياره.

إضافة إلى ذلك فإنّ اعتماد حل البرمجيات المغلقة قد يتطلب القيام بتطوير بعض التجهيزات الحاسوبية الموجودة حالياً لدى الشركة العربية نتيجة لعدم توافقها مع المتطلبات الدنيا لبعض البرمجيات الجديدة، وهو ما سيرفع من تكاليف هذا الحل، أما البرمجيات مفتوحة المصدر فهي لا تتطلب الكثير من التجهيزات الحاسوبية التي ستعمل ضمنها.

إضافةً إلى كل ذلك فإن مبدأ الحرية الذي تبنى عليه البرمجيات مفتوحة المصدر سيضمن للشركة العربية حماية استثماراتها في هذه البرمجيات نتيجةً لتوافر الشيفرة المصدرية لها بما يؤدّي إلى تحرر الشركة من سيطرة الشركات التجارية الإحتكارية.

وفي المقابل فإن اعتماد حل البرمجيات المغلقة يتميز بسهولة أكبر في الإنتقال بالنسبة للمستخدمين نتيجة تشابه واجهة التطبيقات البرمجية الجديدة وآلية عملها مع التطبيقات البرمجية المستخدمة حالياً، ولكن يمكن تجاوز هذه العقبة في اعتماد حل البرمجيات مفتوحة المصدر بتدريب المستخدمين على البرمجيات الجديدة مفتوحة المصدر، وهو ما يعتبر عمليةً سهلةً للغاية خصوصاً مع تطور البرمجيات مفتوحة المصدر وتقارب واجهات استخدامها إلى حد بعيد من البرمجيات المغلقة.

إتخاذ القرار

بعد أسبوعين، قدّم جابر، مدير تقنية المعلومات في الشركة العربية، تقريره لإدارة الشركة، وضمنّه جميع ما توصل إليه من نتائج، وأمام الميزات الواضحة لحل البرمجيات مفتوحة المصدر إضافة إلى ما تقدّمه من وفر هائل في التكاليف (هو ما أسعد بشكل خاص المدير المالي للشركة) كان الخيار باعتماد هذا الحل هو ما اتفق عليه الجميع، ولكن (وكما هي الحال دوماً في التعامل مع الإدارة) كان لديهم بعض التحفظات:

- إنّ اعتماد حل البرمجيات مفتوحة المصدر يتطلب دراسةً مستفيضةً لآلية التحول إلى هذه البرمجيات من البرمجيات المغلقة المعتمدة حالياً لضمان نجاح هذا التحول وتذليل العقبات التي قد تواجه أثناء تنفيذه.
- تمتلك الشركة العربية كماً هائلاً من البيانات المخزنة بصيغ عديدة، منها صيغ خاصة بالبرمجيات المغلقة (وبشكل خاص برمجيات التطبيقات المكتبية)، والشركة غير مستعدة بأي حال من الأحوال للتنازل عن هذه البيانات، لذلك لا بدّ من توضيح إمكانية وآلية التعامل مع هذه البيانات في الحل الجديد.
- تشهد الشركة العربية معدل نمو كبير في أعمالها، هذه الأعمال تتطلب الكثير من التفاعل بين الشركة وزبائنها، وهو ما يفرض تبادل البيانات فيما بينهم، والشركة العربية لا سلطة لها على البيانات الواردة من طرف الزبائن، والتي قد تأتيها بصيغ خاصة بالبرمجيات المغلقة، ولذلك لا بدّ من ضمان إمكانية التعامل مع هذه البيانات في الحل المعتمد.

أجاب جابر (والذي كان مستعداً لكل الإحتمالات) على هذه التحفظات:

- فقد تعهد بإعداد خطّة تفصيلية توضح خطوات التحول إلى المصادر المفتوحة آخذاً بعين الإعتبار جميع العقبات والمصاعب التي قد تواجه هذا التحول.
- كما أوضح أن البرمجيات مفتوحة المصدر التي قام باختيارها تتيح إمكانيات التعامل مع البيانات المخزّنة بصيغ خاصة بالبرمجيات المغلقة (وأشار بالتحديد إلى برمجيات التطبيقات المكتبية حيث أكّد أنّ حزمة التطبيقات المكتبية مفتوحة المصدر التي قام باختيارها OpenOffice تتيح التعامل مع جميع الملفات المخزّنة بصيغ مايكروسوفت أوفيس Microsoft Office وبأن الشرة العربية تملك مطلق الحرية في تحويل جميع هذه الملفات إلى صيغ OpenOffice أو الحفاظ عليها بصيغتها الحالية والتعامل معها وحفظها بنفس هذه الصيغة).
- أشار جابر إلى أنّ التعامل مع الملفات الواردة من الخارج والتي لا سلطة للشركة العربية عليها سيتم بنفس أسلوب التعامل مع البيانات الموجودة حالياً لدى الشركة.

بعد هذه الإيضاحات (وبعد أن تأكدّت الإدارة من أن جابر يعلم تماماً ما الذي سيعمله) لم يكن أمام الإدارة سوى الموافقة على اقتراح مدير تقنية المعلومات باعتماد حل البرمجيات مفتوحة المصدر وإعطائه الضوء الأخضر لمباشرة العمل في تنفيذ عملية التحول إلى هذا الحل.

خطة التحول

بدأ جابر مباشرةً بإعداد خطة التحول إلى البرمجيات مفتوحة المصدر، وبالتوافق مع المنهجية التي طرحناها في فصل سابق من هذا الكتاب فقد تأكد جابر من:

- توفّر التفهم الكامل لأسباب التحوّل قبل البدء.
- التحقق من توفّر دعم فعّال لعملية التحوّل من قبل عناصر تقنية المعلومات إضافة إلى المستخدمين (فقد قام بمحاورة جميع المستخدمين ليوضح لهم القرار الذي اتخذته الإدارة ومدى تأثيره على عمل كل منهم).
- وجود إرادة حقيقية لإنجاز التحول على مستويات الإدارة العليا في المؤسسة.
- بناء المهارات الملائمة في مجال المصادر المفتوحة (عبر توفير الدورات التدريبية اللازمة للمستخدمين).
- قابلية إدارة كل مرحلة من مراحل التحوّل.

ثم قام جابر بوضع خطوات التحول إلى البرمجيات مفتوحة المصدر:

- بدايةً، سيقوم جابر بتحديد البرمجيات مفتوحة المصدر والتي تعمل ضمن بيئة ويندوز Windows وتشغيلها ضمن البيئة الحالية لتعريف المستخدمين بها بشكل أفضل. مجد جابر أنّ جميع برمجيات سطح المكتب التي قام باختيارها تعمل ضمن ويندوز Windows وهي:
    - حزمة التطبيقات المكتبية OpenOffice
    - متصفح الوب Mozilla
- بعد ذلك، سيقوم جابر بإعداد البرمجيات مفتوحة المصدر على المخدّم، وبما أنّ معظم التعديلات التي ستجرى على المخدّم متوافقة بطبيعتها مع أنظمة المستخدمين الحاليّة فإن ذلك سيقلل من الآثار السلبية على أنظمة المستخدمين أثناء عملية

التحوّل. سيقوم جابر بتثبيت البرمجيات التالية على المخدّم وإعدادها بما يتلاءم ومتطلبات الشركة العربية:

- Fedora Core 1
- SendMail
- Apache
- MySQL
- BIND
- DHCPD
- OpenLDAP
- Squid
- Amenda

- ما أن أصبحت البرمجيات مفتوحة المصدر جاهزةً على المخدّم، وبعد أن يتآلف المستخدمين إلى حد ما مع البرمجيات الجديدة، سيصبح بالإمكان تحويل حواسيب المستخدمين بالكامل إلى برمجيات المصادر المفتوحة. هناك عدّة أساليب لتنفيذ هذا التحول:
  - تحويل جميع المستخدمين إلى البرمجيات مفتوحة المصدر دفعةً واحدةً: في نفس اليوم، وهو ما قد يتطلب تنفيذ هذه العملية في يوم عطلة لاستغلال غياب جميع المستخدمين. من ميزات هذا الأسلوب أنه سيلغي الحاجة إلى التفاعل بين المستخدمين في بيئتي المصادر المفتوحة والمغلقة، ومن مساوئه ازدياد احتمالات ظهور المشاكل أثناء التحول وحاجته إلى الكثير من الموارد.
  - تحويل المستخدمين إلى المصادر المفتوحة ضمن مجموعات: حيث يتم نقل المجموعات المرتبطة فيما بينها ضمن سياق العمل (كمجموعة المحاسبة والمالية أو الموارد البشرية مثلاً) في آن واحد، وهو ما سيقلل الحاجة إلى التفاعل بين الأنظمة المختلفة وإحتمالات تعطيل المستخدمين

عن العمل. إن اختيار الحجم الملائم لكل مجموعة سيجعل عملية التحول أكثر قابليةً للإدارة ويقلل من احتياجاتها من الموارد.

- تحويل كل مستخدم على حدة إلى البرمجيات مفتوحة المصدر: وهو نفس الأسلوب السابق ولكن بمجموعات حجم كل منها مستخدم واحد. لا يتطلب هذا الأسلوب الكثير من الموارد لتنفيذ عملية التحول ولكنه فعّال فقط في الشركات الصغيرة.

قرر جابر اعتماد أسلوب التحول ضمن مجموعات بعد أن وجد أنّه الحل الأنسب لحالة الشركة العربية، سيقوم جابر بإعداد البرامج التالية على حواسيب المستخدمين:

- Fedora Core 1
- OpenOffice 1.1
- Mozilla 1.6

تنفيذ خطّة التحول

بعد أن أصبحت خطّة التحول جاهزةً ووافقت عليها إدارة الشركة، قام جابر بتنفيذ هذا التحول بسلاسة دون أن يواجه أية عقبات أو مصاعب كبيرة، وخلال أسبوعين تم تحويل جميع المستخدمين إلى البرمجيات مفتوحة المصدر، كما قام جابر بإجراء دورة تدريبية مبسطة للمستخدمين قام خلالها بتعريفهم بمبادئ البرمجيات مفتوحة المصدر وكيفية التعامل مع نظام التشغيل لينكس وحزمة التطبيقات المكتبية OpenOffice، أثناء هذه الدورة التي استغرقت أسبوعاً واحداً أعرب المستخدمون أنهم مرتاحون في التعامل مع البرمجيات الجديدة لأن واجهة استخدامها تشبه إلى حد بعيد البرمجيات المغلقة التي اعتادوا عليها،

(وهو ما طمأن جابر حيث أنّ أكثر ما كان يخشاه أثناء تنفيذ عملية التحول هو استياء المستخدمين ومعارضتهم لاستخدام البرمجيات الجديدة).

حالياً، تستخدم الشركة العربية البرمجيات مفتوحة المصدر في جميع نشاطاتها، وقد قام عدد من الموظفين لديها بتطوير بعض هذه البرمجيات لتلبية متطلبات العمل المتزايدة في الشركة، كما دفع نجاح الشركة العربية في اعتماد البرمجيات مفتوحة المصدر عدداً من زبائنها إلى استخدام هذه البرمجيات في أعمالهم أيضاً.

كما قررت إدارة الشركة منح جابر، مهندس عملية التحول إلى المصادر المفتوحة مكافأةً سخيةً وأهدته حاسوباً محمولاً يعمل بنظام التشغيل لينكس.

أسئلة عليك إجابتها

سواء كنت تقوم ببناء نظام تقنية المعلومات لمؤسستك أو لأحد زبائنك من البداية أو تبحث في إمكانيات تطوير نظام قائم عليك أن تعتبر أنّ برمجيات المصادر المفتوحة هي أحد الخيارات المتاحة لديك، ولكي تقوم بتحديد مدى ملاءمة هذه البرمجيات لوضع مؤسستك حاول الإجابة على الأسئلة التالية:

- ما هي المتطلبات الأساسية للمؤسسة من أنظمة تقنية المعلومات، وبشكل خاص البرمجيات المستخدمة في هذه الأنظمة؟
- ما هي الخيارات المختلفة المتاحة لتلبية هذه المتطلبات سواء كانت مغلقةً أم مفتوحة المصدر؟

- ما هي ميزات ومساوئ كل من هذه الحلول؟
- ما هي التكاليف المترتبة على اختيار أي من الحلول المتاحة؟
- ما هي المكاسب التي ستجنيها المؤسسة من اعتماد المصادر المفتوحة؟
- كيف ستقوم بتبرير الحل الذي قمت باختياره لإدارة المؤسسة؟
- ما هي متطلبات التفاعل والتوافقية بين البرمجيات المفتوحة والمغلقة في المؤسسة؟
- ما هي العوائق التي قد تواجه أية عملية تحول أثناء تنفيذها؟